COUNTERFACTUAL TRAINING: TEACHING MODELS PLAUSIBLE AND ACTIONABLE EXPLANATIONS

A PREPRINT

Patrick Altmeyer
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

p.altmeyer@tudelft.nl

Aleksander Buszydlik
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Arie van Deursen
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Cynthia C. S. Liem
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

July 22, 2025

ABSTRACT

We propose a novel training regime termed counterfactual training that leverages counterfactual explanations to increase the explanatory capacity of models. Counterfactual explanations have emerged as a popular post-hoc explanation method for opaque machine learning models: they inform how factual inputs would need to change in order for a model to produce some desired output. To be useful in real-word decision-making systems, counterfactuals should be plausible with respect to the underlying data and actionable with respect to the feature mutability constraints. Much existing research has therefore focused on developing post-hoc methods to generate counterfactuals that meet these desiderata. In this work, we instead hold models directly accountable for the desired end goal: counterfactual training employs counterfactuals during the training phase to minimize the divergence between learned representations and plausible, actionable explanations. We demonstrate empirically and theoretically that our proposed method facilitates training models that deliver inherently desirable counterfactual explanations and exhibit greatly improved adversarial robustness.

Keywords Counterfactual Training • Counterfactual Explanations • Algorithmic Recourse • Explainable AI • Representation Learning

1 Introduction

Today's prominence of artificial intelligence (AI) has largely been driven by the success of representation learning with high degrees of freedom: instead of relying on features and rules hand-crafted by humans, modern machine

learning (ML) models are tasked with learning highly complex representations directly from the data, guided by narrow objectives such as predictive accuracy (Goodfellow, Bengio, and Courville 2016). These models tend to be so complex that humans cannot easily interpret their decision logic.

Counterfactual explanations (CE) have become a key part of the broader explainable AI (XAI) toolkit (Molnar 2022) that can be applied to make sense of this complexity. Originally proposed by Wachter, Mittelstadt, and Russell (2017), CEs prescribe minimal changes for factual inputs that, if implemented, would prompt some fitted model to produce an alternative, more desirable output. This is useful and necessary to not only understand how opaque models make their predictions, but also to provide algorithmic recourse (AR) to individuals subjected to them: a retail bank, for example, could use CE to provide meaningful feedback to unsuccessful loan applicants that were rejected based on an opaque automated decision-making (ADM) system (Figure 1).

For such feedback to be meaningful, counterfactual explanations need to fulfill certain desiderata (Verma et al. 2022; Karimi et al. 2021)—they should be faithful to the model (Altmeyer et al. 2024), plausible (Joshi et al. 2019) and actionable (Ustun, Spangher, and Liu 2019). Plausibility is typically understood as counterfactuals being *in-domain*: unsuccessful loan applicants that implement the provided recourse should end up with credit profiles that are genuinely similar to that of individuals who have successfully repaid their loans in the past. Actionable explanations comply with practical constraints: a young, unsuccessful loan applicant cannot increase their age in an instance.

Existing state-of-the-art (SOTA) approaches in the field have largely focused on designing model-agnostic CE methods that identify subsets of counterfactuals, which comply with specific desiderata. This is problematic, because the narrow focus on any specific desideratum can adversely affect others: it is possible, for example, to generate plausible counterfactuals for models that are also highly vulnerable to implausible, possibly adversarial counterfactuals (Altmeyer et al. 2024). In this work, we therefore embrace the paradigm that models (as opposed to explanation methods) should be held accountable for explanations that are plausible and actionable. While previous work has shown that at least plausibility can be indirectly achieved through existing techniques aimed at models' generative capacity, generalization and robustness (Altmeyer et al. 2024; Augustin, Meinke, and Hein 2020; Schut et al. 2021), we directly incorporate both plausibility and actionability in the training objective of models to improve their overall explanatory capacity.

Specifically, we propose **counterfactual training (CT)**: a novel training regime that leverages counterfactual explanations on-the-fly to ensure that differentiable models learn plausible and actionable explanations for the underlying data, while at the same time also being more robust to adversarial examples (AE). Figure 1 illustrates the outcomes of CT compared to a conventionally trained model. First, in panel (a), faithful and valid counterfactuals end up near the decision boundary forming a clearly distinguishable cluster in the target class (orange). In panel (b), CT is applied to the same underlying linear classifier architecture resulting in much more plausible counterfactuals. In panel (c), the classifier is again trained conventionally and we have introduced a mutability constraint on the *age* feature at test time—counterfactuals are valid but the classifier is roughly equally sensitive to both features. By contrast, the decision boundary in panel (d) has titled, making the model trained with CT relatively less sensitive to the immutable *age* feature. To achieve these outcomes, CT draws inspiration from the literature on contrastive and robust learning: we contrast faithful CEs with ground-truth data while protecting immutable features, and capitalize on methodological links between CE and AE by penalizing the model's adversarial loss on interim (*nascent*) counterfactuals. To the best of our knowledge, CT represents the first venture in this direction with promising empirical and theoretical results.

The remainder of this manuscript is structured as follows. Section 2 presents related work, focusing on the links to contrastive and robust learning. Then follow our two principal contributions. In Section 3, we introduce our methodological framework and show theoretically that it can be employed to respect global actionability constraints. In our experiments (Section 4), we find that thanks to counterfactual training, (1) the implausibility of CEs decreases by up to 90%; (2) the cost of reaching valid counterfactuals with protected features decreases by 19% on average; and (3) models' adversarial robustness improves across the board. Finally, we discuss open challenges in Section 5 and conclude in Section 6.

2 Related Literature

To make the desiderata for our framework more concrete, we follow previous work in tying the explanatory capacity of models to the quality of CEs that can be generated for them (Altmeyer et al. 2024; Augustin, Meinke, and Hein 2020). For simplicity, we refer to "explanatory capacity" as "explainability" in the rest of this manuscript (see Def. 3.1).

2.1 Explainability and Contrastive Learning

In a closely related work, Altmeyer et al. (2024) show that model averaging and, in particular, contrastive model objectives can produce more explainable and hence trustworthy models. The authors propose a way to generate coun-

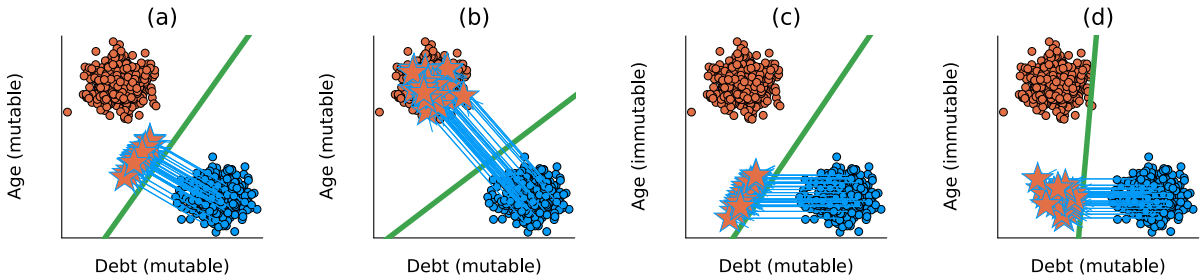

Figure 1: Counterfactual explanations (stars) for linear classifiers trained under different regimes on synthetic data: (a) conventional training, all mutable; (b) CT, all mutable; (c) conventional, *age* immutable; (d) CT, *age* immutable. The linear decision boundary is shown in green along with training data colored according to ground-truth labels: y^- = "loan withheld" (blue) and y^+ = "loan provided" (orange). Class and feature annotations (*debt* and *age*) are for illustrative purposes.

terfactuals that are maximally faithful in that they are consistent with what models have learned about the underlying data. Formally, they rely on tools from energy-based modelling (Teh et al. 2003) to minimize the contrastive divergence between the distribution of counterfactuals and the conditional posterior over inputs learned by a model. Their algorithm, *ECCCo*, yields plausible counterfactual explanations if and only if the underlying model has learned representations that align with them. The authors find that both deep ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) and joint energy-based models (JEMs) (Grathwohl et al. 2020), a form of constrastive learning, tend to do well in this regard.

It helps to look at these findings through the lens of representation learning with high degrees of freedom. Deep ensembles are approximate Bayesian model averages, which are particularly effective when models are underspecified by the available data (Wilson 2020). Averaging across solutions mitigates the risk of overrelying on a single locally optimal representation that corresponds to semantically meaningless explanations. Likewise, previous work of Schut et al. (2021) found that generating plausible ("interpretable") CEs is almost trivial for deep ensembles that have undergone adversarial training. The case for JEMs is even clearer: they optimize a hybrid objective that induces both high predictive performance and strong generative capacity (Grathwohl et al. 2020), which resembles the idea of aligning models with plausible explanations and has inspired CT.

2.2 Explainability and Robust Learning

Augustin, Meinke, and Hein (2020) show that CEs tend to be more meaningful ("explainable") if the underlying model is more robust to adversarial examples. Once again, we can make intuitive sense of this finding if we look at adversarial training (AT) through the lens of representation learning with high degrees of freedom: highly complex and flexible models may learn representations that make them sensitive to implausible or even adversarial examples (Szegedy et al. 2014). Thus, by inducing models to "unlearn" susceptibility to such examples, adversarial training can effectively remove implausible explanations from the solution space.

This interpretation of the link between explainability through counterfactuals on the one side, and robustness to adversarial examples on the other is backed by empirical evidence. Sauer and Geiger (2021) demonstrate that using counterfactual images during classifier training improves model robustness. Similarly, Abbasnejad et al. (2020) argue that counterfactuals represent potentially useful training data in machine learning, especially in supervised settings where inputs may be reasonably mapped to multiple outputs. They, too, show that augmenting the training data of (image) classifiers can improve generalization performance. Finally, Teney, Abbasnedjad, and Hengel (2020) argue that counterfactual pairs tend to exist in training data. Hence, their approach aims to identify similar input samples with different annotations and ensure that the gradient of the classifier aligns with the vector between such pairs of counterfactual inputs using a cosine distance loss function.

CEs have also been used to improve models in the natural language processing domain. For example, Wu et al. (2021) propose *Polyjuice*, a general-purpose CE generator for language models and demonstrate that the augmentation of training data with *Polyjuice* improves robustness in a number of tasks, while Luu and Inoue (2023) introduce the *Counterfactual Adversarial Training* (CAT) framework that aims to improve generalization and robustness of language models by generating counterfactuals for training samples that are subject to high predictive uncertainty.

There have also been several attempts at formalizing the relationship between counterfactual explanations and adversarial examples. Pointing to clear similarities in how CEs and AEs are generated, Freiesleben (2022) makes the case for jointly studying the opaqueness and robustness problems in representation learning. Formally, AEs can be seen as

the subset of CEs for which misclassification is achieved (Freiesleben 2022). Similarly, Pawelczyk et al. (2022) show that CEs and AEs are equivalent under certain conditions.

Two other works are closely related to ours in that they use counterfactuals during training with the explicit goal of affecting certain properties of the post-hoc counterfactual explanations. Firstly, Ross, Lakkaraju, and Bastani (2024) propose a way to train models that guarantee recourse to a positive target class with high probability. Their approach builds on adversarial training by explicitly inducing susceptibility to targeted AEs for the positive class. Additionally, the method allows for imposing a set of actionability constraints ex-ante. For example, users can specify that certain features are immutable. Secondly, Guo, Nguyen, and Yadav (2023) are the first to propose an end-to-end training pipeline that includes CEs as part of the training procedure. Their *CounterNet* network architecture includes a predictor and a CE generator, where the parameters of the CE generator are learnable. Counterfactuals are generated during each training iteration and fed back to the predictor. In contrast, we impose no restrictions on the ANN architecture at all.

3 Counterfactual Training

This section introduces the counterfactual training framework, applying ideas from contrastive and robust learning to counterfactual explanations. CT produces models whose learned representations align with plausible explanations that comply with user-defined actionability constraints.

Counterfactual explanations are typically generated by solving variations of the following optimization problem,

$$\min_{\mathbf{x}' \in \mathcal{X}^D} \{\text{yloss}(\mathbf{M}_\theta(\mathbf{x}'), \mathbf{y}^+) + \lambda \text{reg}(\mathbf{x}')\} \tag{1}$$

where $\mathbf{M}_\theta : \mathcal{X} \mapsto \mathcal{Y}$ denotes a classifier, $\mathbf{x}'$ denotes the counterfactual with D features and $\mathbf{y}^+ \in \mathcal{Y}$ denotes some target class. The $\text{yloss}(\cdot)$ function quantifies the discrepancy between current model predictions for $\mathbf{x}'$ and the target class (a conventional choice is cross-entropy). Finally, we use $\text{reg}(\cdot)$ to denote any form of regularization used to induce certain properties on the counterfactual. In their seminal paper, Wachter, Mittelstadt, and Russell (2017) propose regularizing the distance between counterfactuals and their original factual values to ensure that individuals seeking recourse through CE face minimal costs in terms of feature changes. Different variations of Equation 1 have been proposed in the literature to address many desiderata including the ones discussed above (faithfulness, plausibility and actionability). Like Wachter, Mittelstadt, and Russell (2017), most of these approaches rely on gradient descent to optimize Equation 1. For more details on the approaches tested in this work, we refer the reader to the supplementary appendix. In the following, we describe in detail how counterfactuals are generated and used in counterfactual training.

3.1 Proposed Training Objective

The goal of CT is to improve model explainability by aligning models with faithful explanations that are plausible and actionable. Formally, we define explainability as follows:

Definition 3.1 (Model Explainability)**.** Let $\mathbf{M}_\theta : \mathcal{X} \mapsto \mathcal{Y}$ denote a supervised classification model that maps from the D-dimensional input space $\mathcal{X}$ to representations $\phi(\mathbf{x}; \theta)$ and finally to the K-dimensional output space $\mathcal{Y}$. Assume that for any given input-output pair $\{\mathbf{x}, \mathbf{y}\}_i$ there exists a counterfactual $\mathbf{x}' = \mathbf{x} + \Delta : \mathbf{M}_\theta(\mathbf{x}') = \mathbf{y}^+ \neq \mathbf{y} = \mathbf{M}_\theta(\mathbf{x})$, where $\arg\max_y \mathbf{y}^+ = y^+$ is the index of the target class.

We say that $\mathbf{M}_\theta$ has an **explanatory capacity** to the extent that faithfully generated, valid counterfactuals are also plausible and actionable. We define these properties as:

- (Faithfulness) $\int^A p_\theta(\mathbf{x}'|\mathbf{y}^+) d\mathbf{x} \to 1$; A is an arbitrarily small region around $\mathbf{x}'$.
- (Plausibility) $\int^A p(\mathbf{x}'|\mathbf{y}^+) d\mathbf{x} \to 1$; A as specified above.
- (Actionability) Perturbations Δ may be subject to some actionability constraints.

Here, $p_\theta(\mathbf{x}|\mathbf{y}^+)$ denotes the conditional posterior distribution over inputs. For simplicity, we refer to a model with high explanatory capacity as **explainable** in this manuscript.

The characterization of faithfulness and plausibility in Def. 3.1 follows Altmeyer et al. (2024), with adapted notation. Intuitively, plausible counterfactuals are consistent with the data and faithful counterfactuals are consistent with what the model has learned about the input data. Ac tionability constraints in Def. 3.1 vary and depend on the context in which $\mathbf{M}_\theta$ is deployed. In this work, we choose to only consider domain and mutability constraints for individual features x_d for $d = 1, ..., D$. We also limit ourselves to classification tasks for reasons discussed in Section 5.

Let $\mathbf{x}'_t$ for $t = 0, ..., T$ denote a counterfactual generated through gradient descent over T iterations as originally proposed by Wachter, Mittelstadt, and Russell (2017). CT adopts gradient-based CE search in training to generate

on-the-fly model explanations $\mathbf{x}'$ for the training samples. We use the term *nascent* to denote interim counterfactuals $\mathbf{x}'_{t\leq T}$ that have not yet converged. As we explain below, these nascent counterfactuals can be stored and repurposed as adversarial examples. Conversely, we consider counterfactuals $\mathbf{x}'_T$ as *mature* explanations if they have either exhausted all T iterations or converged by reaching a pre-specified threshold, τ, for the predicted probability of the target class: $\mathcal{S}(\mathbf{M}_\theta(\mathbf{x}'))[y^+] \geq \tau$, where $\mathcal{S}$ is the softmax function.

Formally, we propose the following counterfactual training objective to train explainable (as in Def. 3.1) models,

$$\begin{aligned} &\min_\theta \text{yloss}(\mathbf{M}_\theta(\mathbf{x}), \mathbf{y}) + \lambda_{\text{div}}\text{div}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) \\ &+\lambda_{\text{adv}}\text{advloss}(\mathbf{M}_\theta(\mathbf{x}'_{t\leq T}), \mathbf{y}) + \lambda_{\text{reg}}\text{ridge}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) \end{aligned} \tag{2}$$

where $\text{yloss}(\cdot)$ is any classification loss that induces discriminative performance (e.g., cross-entropy). The second and third terms are explained in detail below. For now, they can be summarized as inducing explainability directly and indirectly by penalizing (1) the contrastive divergence, $\text{div}(\cdot)$, between mature counterfactuals $\mathbf{x}'_T$ and observed samples $\mathbf{x}^+ \in \mathcal{X}^+ = \{\mathbf{x} : y = y^+\}$ in the target class y^+, and (2) the adversarial loss, $\text{advloss}(.)$, wrt. nascent counterfactuals $\mathbf{x}'_{t\leq T}$. Finally, $\text{ridge}(\cdot)$ denotes a Ridge penalty (ℓ_2-norm) that regularizes the magnitude of the energy terms involved in $\text{div}(\cdot)$ (Du and Mordatch 2020). The trade-offs between these components are adjusted through λ_{div}, λ_{adv} and λ_{reg}. The full training regime is sketched out in Algorithm 1.

Algorithm 1 Counterfactual Training

Require: Training dataset $\mathcal{D}$, initialize model $\mathbf{M}_\theta$
1: **while** not converged **do**
2: Sample $\mathbf{x}$ and $\mathbf{y}$ from dataset $\mathcal{D}$.
3: Sample $\mathbf{x}'_0$, $\mathbf{y}^+$ and $\mathbf{x}^+$.
4: **for** $t = 1$ to T **do**
5: Backpropagate $\nabla_{\mathbf{x}'}$ through Equation 1. Store $\mathbf{x}'_t$.
6: **end for**
7: Backpropagate ∇_θ through Equation 2.
8: **end while**
9: **return** $\mathbf{M}_\theta$

3.2 Directly Inducing Explainability with Contrastive Divergence

Grathwohl et al. (2020) observe that any classifier can be re-interpreted as a joint energy-based model that learns to discriminate output classes conditional on the observed (training) samples from $p(\mathbf{x})$ and the generated samples from $p_\theta(\mathbf{x})$. The authors show that JEMs can be trained to perform well at both tasks by directly maximizing the joint log-likelihood: $\log p_\theta(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{y}|\mathbf{x}) + \log p_\theta(\mathbf{x})$, where the first term can be optimized using cross-entropy as in Equation 2. To optimize $\log p_\theta(\mathbf{x})$, they minimize the contrastive divergence between the observed samples from $p(\mathbf{x})$ and samples generated from $p_\theta(\mathbf{x})$.

To generate samples, Grathwohl et al. (2020) use Stochastic Gradient Langevin Dynamics (SGLD) with an uninformative prior for initialization but we depart from their methodology: we propose to leverage counterfactual explainers to generate counterfactuals of observed training samples. Specifically, we have:

$$\text{div}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) = \mathcal{E}_\theta(\mathbf{x}^+, y) - \mathcal{E}_\theta(\mathbf{x}'_T, y) \tag{3}$$

where $\mathcal{E}_\theta(\cdot)$ denotes the energy function defined as $\mathcal{E}_\theta(\mathbf{x}, y) = -\mathbf{M}_\theta(\mathbf{x})[y^+]$, with y^+ denoting the index of the randomly drawn target class, $y^+ \sim p(y)$. Conditional on the target class y^+, $\mathbf{x}'_T$ denotes a mature counterfactual for a randomly sampled factual from a non-target class generated with a gradient-based CE generator for up to T iterations. Intuitively, the gradient of Equation 3 decreases the energy of observed training samples (positive samples) while increasing the energy of counterfactuals (negative samples) (Du and Mordatch 2020). As the counterfactuals get more plausible (Def. 3.1) during training, these opposing effects gradually balance each other out (Lippe 2024).

Since maturity of counterfactuals in terms of a probability threshold is often reached before T, this form of sampling is not only more closely aligned with Def. 3.1., but can also speed up training times compared to SGLD. The departure from SGLD also allows us to tap into the vast repertoire of explainers that have been proposed in the literature to meet different desiderata. For example, many methods support domain and mutability constraints. In principle, any existing approach for generating CEs is viable, so long as it does not violate the faithfulness condition. Like JEMs (Murphy 2022), counterfactual training can be considered a form of contrastive representation learning.

3.3 Indirectly Inducing Explainability with Adversarial Robustness

Based on our analysis in Section 2, counterfactuals $\mathbf{x}'$ can be repurposed as additional training samples (Balashankar et al. 2023; Luu and Inoue 2023) or adversarial examples (Freiesleben 2022; Pawelczyk et al. 2022). This leaves some flexibility with regards to the choice for the advloss$(\cdot)$ term in Equation 2. An intuitive functional form, but likely not the only sensible choice, is inspired by adversarial training:

$$\begin{aligned} \text{advloss}(\mathbf{M}_\theta(\mathbf{x}'_{t\leq T}), \mathbf{y}; \varepsilon) &= \text{yloss}(\mathbf{M}_\theta(\mathbf{x}'_{t_\varepsilon}), \mathbf{y}) \\ t_\varepsilon &= \max_t \{t : ||\Delta_t||_\infty < \varepsilon\} \end{aligned} \tag{4}$$

Under this choice, we consider nascent counterfactuals $\mathbf{x}'_{t\leq T}$ as AEs as long as the magnitude of the perturbation to any single feature is at most ε. This is closely aligned with Szegedy et al. (2014) who define an adversarial attack as an "imperceptible non-random perturbation". Thus, we work with a different distinction between CE and AE than Freiesleben (2022) who considers misclassification as the distinguishing feature of adversarial examples. One of the key observations of this work is that we can leverage CEs during training and get AEs essentially for free to reap the aforementioned benefits of adversarial training.

3.4 Encoding Actionability Constraints

Many existing counterfactual explainers support domain and mutability constraints. In fact, both types of constraints can be implemented for any explainer that relies on gradient descent in the feature space for optimization (Altmeyer, Deursen, and Liem 2023). In this context, domain constraints can be imposed by simply projecting counterfactuals back to the specified domain, if the previous gradient step resulted in updated feature values that were out-of-domain. Similarly, mutability constraints can be enforced by setting partial derivatives to zero to ensure that features are only perturbed in the allowed direction, if at all.

Since actionability constraints are binding at test time, we also impose them when generating $\mathbf{x}'$ during each training iteration to inform model representations. Through their effect on $\mathbf{x}'$, both types of constraints influence model outcomes via Equation 3. Here it is crucial that we avoid penalizing implausibility that arises due to mutability constraints. For any mutability-constrained feature d this can be achieved by enforcing $\mathbf{x}^+[d] - \mathbf{x}'[d] := 0$ whenever perturbing $\mathbf{x}'[d]$ in the direction of $\mathbf{x}^+[d]$ would violate mutability constraints. Specifically, we set $\mathbf{x}^+[d] := \mathbf{x}'[d]$ if:

1. Feature d is strictly immutable in practice.
2. $\mathbf{x}^+[d] > \mathbf{x}'[d]$, but d can only be decreased in practice.
3. $\mathbf{x}^+[d] < \mathbf{x}'[d]$, but d can only be increased in practice.

From a Bayesian perspective, setting $\mathbf{x}^+[d] := \mathbf{x}'[d]$ can be understood as assuming a point mass prior for $p(\mathbf{x}^+)$ wrt. feature d. Intuitively, we think of this as ignoring implausibility costs of immutable features, which effectively forces the model to instead seek plausibility through the remaining features. This can be expected to result in relatively lower sensitivity to immutable features; and higher relative sensitivity to mutable features should make mutability-constrained recourse less costly (Section 4). Under certain conditions, this result holds theoretically; for the proof, see the supplementary appendix:

Proposition 3.1 (Protecting Immutable Features)**.** *Let $f_\theta(\mathbf{x}) = \mathcal{S}(\mathbf{M}_\theta(\mathbf{x})) = \mathcal{S}(\Theta\mathbf{x})$ denote a linear classifier with softmax activation $\mathcal{S}$ where $y \in \{1, ..., K\} = \mathcal{K}$ and $\mathbf{x} \in \mathbb{R}^D$. Assume multivariate Gaussian class densities with common diagonal covariance matrix $\Sigma_k = \Sigma$ for all $k \in \mathcal{K}$, then protecting an immutable feature from the contrastive divergence penalty will result in lower classifier sensitivity to that feature relative to the remaining features, provided that at least one of those is discriminative and mutable.*

4 Experiments

We seek to answer the following four research questions:

(RQ1) To what extent does the CT objective in Equation 1 induce models to learn plausible explanations?

(RQ2) To what extent does CT result in more favorable algorithmic recourse outcomes in the presence of actionability constraints?

(RQ3) To what extent does CT influence the adversarial robustness of trained models?

(RQ4) What are the effects of hyperparameter selection on counterfactual training?

4.1 Experimental Setup

Our focus is the improvement in explainability (Def. 3.1). Thus, we primarily look at the plausibility and cost of faithfully generated counterfactuals at test time. Other metrics, such as validity and redundancy, are reported in the supplementary appendix. To measure the cost, we follow the standard proxy of distances (ℓ_1-norm) between factuals and counterfactuals. For plausibility, we assess how similar CEs are to observed samples in the target domain, $\mathbf{X}^+ \subset \mathcal{X}^+$. We rely on the metric used by Altmeyer et al. (2024),

$$\text{IP}(\mathbf{x}', \mathbf{X}^+) = \frac{1}{|\mathbf{X}^+|} \sum_{\mathbf{x} \in \mathbf{X}^+} \text{dist}(\mathbf{x}', \mathbf{x}) \tag{5}$$

and introduce a novel divergence-based adaptation,

$$\text{IP}^*(\mathbf{X}', \mathbf{X}^+) = \text{MMD}(\mathbf{X}', \mathbf{X}^+) \tag{6}$$

where $\mathbf{X}'$ denotes a collection of counterfactuals and $\text{MMD}(\cdot)$ is the unbiased estimate of the squared population maximum mean discrepancy, proposed by Gretton et al. (2012). The metric in Equation 6 is equal to zero if and only if the two distributions are exactly the same, $\mathbf{X}' = \mathbf{X}^+$.

To assess outcomes with respect to actionability for non-linear models, we look at the average costs of valid counterfactuals in terms of their distances from factual starting points. While this an imperfect proxy of sensitivity, we hypothesize that CT can reduce these costs by teaching models to seek plausibility with respect to mutable features, much like we observe in Figure 1 in panel (d) compared to (c). We supplement this analysis with qualitative findings for integrated gradients (Sundararajan, Taly, and Yan 2017). Finally, for predictive performance, we use standard metrics, such as robust accuracy estimated on adversarially perturbed data using FGSM (Goodfellow, Shlens, and Szegedy 2015).

We run experiments with three gradient-based generators: *Generic* of Wachter, Mittelstadt, and Russell (2017) as a simple baseline approach, *REVISE* (Joshi et al. 2019) that aims to generate plausible counterfactuals using a surrogate Variational Autoencoder (VAE), and *ECCCo* (Altmeyer et al. 2024), which targets faithfulness.

We make use of nine classification datasets common in the CE/AR literature. Four of them are synthetic with two classes and different characteristics: linearly separable clusters (*LS*), overlapping clusters (*OL*), concentric circles (*Circ*), and interlocking moons (*Moon*). Next, we have four real-world binary tabular datasets: *Adult* (Census data) of Becker and Kohavi (1996), California housing (*CH*) of Pace and Barry (1997), Default of Credit Card Clients (*Cred*) of Yeh (2016), and Give Me Some Credit (*GMSC*) from Kaggle (2011). Finally, for the convenience of illustration, we use the 10-class *MNIST* (LeCun 1998).

To assess CT, we investigate the improvements in performance metrics when using it on top of a weak baseline (BL): a multilayer perceptron (*MLP*). This is the best way to get a clear picture of the effectiveness of CT, and it is consistent with evaluation practices in the related literature (Goodfellow, Shlens, and Szegedy 2015; Ross, Lakkaraju, and Bastani 2024; Teney, Abbasnedjad, and Hengel 2020).

4.2 Experimental Results

Our main results for plausibility and actionability for *MLP* models are summarised in Table 1 that presents counterfactual outcomes grouped by dataset along with standard errors averaged across bootstrap samples. Asterisks (*) are used when the bootstrapped 99%-confidence interval of differences in mean outcomes does *not* include zero, so the observed effects are statistically significant at the 0.01 level.

The first two columns (IP and IP^*) show the percentage reduction in implausibility for our two metrics when using CT on top of the weak baseline. As an example, consider the first row for *LS* data: the observed positive values indicate that faithful counterfactuals are around 30-55% more plausible for models trained with CT, in line with our observations in panel (b) of Figure 1 compared to panel (a).

The third column shows the results for a scenario when mutability constraints are imposed on the selected features. Again, we are comparing CT to the baseline, so reductions in the positive direction imply that valid counterfactuals are "cheaper" (more actionable) when using CT with feature protection. Relating this back to Figure 1, the third column represents the reduction in distances travelled by counterfactuals in panel (d) compared to panel (c). In the following paragraphs, we summarize the results for all datasets.

4.2.1 Plausibility (RQ1).

CT generally produces substantial and statistically significant improvements in plausibility.

Average reductions in IP range from around 7% for *MNIST* to almost 60% for *Circ*. For the real-world tabular datasets they are around 12% for *CH* and *Cred* and almost 25% for *GMSC*; for *Adult* and *OL* we find no significant impact

Table 1: Key evaluation metrics for valid counterfactual along with bootstrapped standard errors for all datasets. **Plausibility** (columns 1-2): percentage reduction in implausibility for IP and IP*, respectively; **Cost / Actionability** (column 3): percentage reduction in costs when selected features are protected. Outcomes are aggregated across bootstrap samples (100 rounds) and varying degrees of the energy penalty λ_{egy} used for *ECCCo* at test time. Asterisks (*) indicate that the bootstrapped 99%-confidence interval of differences in mean outcomes does *not* include zero.

Data	**IP** $(-\%)$	**IP*** $(-\%)$	**Cost** $(-\%)$
LS	$29.05 \pm 0.67^*$	$55.33 \pm 2.03^*$	$14.07 \pm 0.60^*$
Circ	$56.29 \pm 0.44^*$	$89.38 \pm 9.30^*$	$45.55 \pm 0.76^*$
Moon	$20.62 \pm 0.69^*$	$19.26 \pm 8.12^*$	$2.86 \pm 1.03^*$
OL	-1.13 ± 0.88	-24.52 ± 14.52	$38.39 \pm 2.21^*$
Adult	0.77 ± 1.34	$32.29 \pm 6.87^*$	-2.82 ± 4.88
CH	$12.05 \pm 1.41^*$	$70.27 \pm 3.72^*$	$40.71 \pm 1.55^*$
Cred	$12.31 \pm 1.84^*$	$54.89 \pm 11.21^*$	$-17.43 \pm 5.17^*$
GMSC	$23.44 \pm 1.99^*$	$73.31 \pm 4.83^*$	$62.64 \pm 2.04^*$
MNIST	$7.05 \pm 1.80^*$	-25.09 ± 109.05	-12.34 ± 6.52
Avg.	17.83	38.35	19.07

of CT on IP. Reductions in IP* are even more substantial and generally statistically significant, although the average degree of uncertainty is higher than for IP: reductions range from around 20% (*Moons*) to almost 90% (*Circ*). The only negative findings are for OL and MNIST, but they are not statistically significant. A qualitative inspection of the counterfactuals in Figure 4 (columns 2-5) suggests recognizable digits 1-4 for the model trained with CT (bottom row), unlike the baseline (top row).

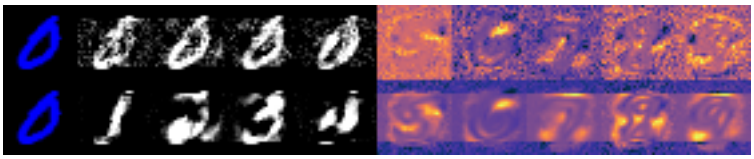

Figure 2: Visual explanations for *MNIST* for BL (top) and CT (bottom). **Plausibility**: col. 1 is a random factual 0 (blue); cols. 2-5 are corresponding *ECCCo* counterfactuals in target classes 1 to 4. **Actionability**: cols. 6-10 show integrated gradients averaged over test images in classes 5 to 9.

4.2.2 Actionability (RQ2).

CT tends to improve actionability in the presence of immutable features, but this is not guaranteed if the assumptions in Proposition 3.1 are violated.

For synthetic datasets, we always protect the first feature; for all real-world tabular datasets we could identify and protect an *age* variable; for *MNIST*, we protect the five upper and lower pixel rows of the full image. Statistically significant reductions in costs overwhelmingly point in the expected positive direction reaching up to around 60% for *GMSC*. Only in the case of *Cred*, average costs increase, likely because any potential benefits from protecting the *age* are outweighed by the increase in costs required for greater plausibility. The findings for *Adult* and *MNIST* are not significant. A qualitative inspection of the class-conditional integrated gradients in Figure 4 (columns 6-10) suggests that CT still has the expected effect: the model (bottom) is insensitive (blue) to the protected rows of pixels; details of this experiment are reported in the supplementary appendix.

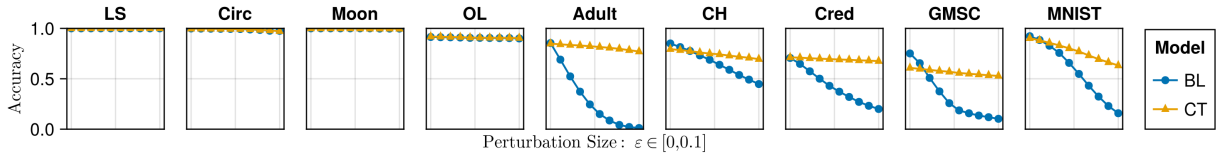

Figure 3: Test accuracies on adversarially perturbed data with varying perturbation sizes for all non-synthetic datasets.

4.2.3 Predictive Performance (RQ3).

Models trained with CT are substantially more robust to gradient-based adversarial attacks than conventionally-trained baselines.

Test accuracies on adversarially perturbed data are shown in Figure 3. The perturbations size, $\varepsilon \in [0, 0.1]$, increases along the horizontal axis and includes zero, corresponding to standard test accuracy for non-perturbed data. For all synthetic datasets, predictive performance of CT is virtually identical to the baseline and unaffected by perturbations. For all real-world datasets, we find that CT substantially improves robustness: while in some cases baseline accuracies drop to essentially zero for large enough perturbation sizes, accuracies of CT models remain remarkably robust.

4.2.4 Hyperparameter settings (RQ4).

CT is highly sensitive to the choice of a CE generator and its hyperparameters but (1) we observe manageable patterns, and (2) we can usually identify settings that improve either plausibility or actionability, and typically both of them at the same time.

We evaluate the impacts of three types of hyperparameters on CT. In this section we focus on the highlights and make the full results available in the supplementary appendix.

Firstly, we find that optimal results are generally obtained when using *ECCCo* to generate counterfactuals. Conversely, using a generator that may inhibit faithfulness (*REVISE*), tends to yield poor results. Concerning hyperparameters that guide the gradient-based counterfactual search, we find that increasing T, the maximum number of steps, generally yields better outcomes because more CEs can mature. Relatedly, we also find that the effectiveness and stability of CT is positively associated with the total number of counterfactuals generated during each training epoch. The impact of τ, the decision threshold, is more difficult to predict. On "harder" datasets it may be difficult to satisfy high τ for any given sample (i.e., also factuals) and so increasing this threshold does not seem to correlate with better outcomes. In fact, $\tau = 0.5$ generally leads to optimal results as it is associated with high proportions of mature counterfactuals.

Secondly, the strength of the energy regularization, λ_{reg} is highly impactful and should be set sufficiently high to avoid common problems associated with exploding gradients. The sensitivity with respect to λ_{div} and λ_{adv} is much less evident. While high values of λ_{reg} may increase the variability in outcomes when combined with high values of λ_{div} or λ_{adv}, this effect is not particularly pronounced.

Finally, we also observe desired improvements when CT was combined with conventional training and applied only for the final 50% of epochs of the complete training process. Put differently, CT can improve the explainability of models in a post-hoc, fine-tuning manner.

5 Discussion

As our results indicate, counterfactual training produces models that are more explainable. Nonetheless, these advantages come at the cost of two important limitations.

Interventions on features have implications for fairness. We provide a method to modify the sensitivity of a model to certain features, which can be misused by enforcing explanations based on features that are more difficult to modify by a (group of) decision subjects. Such abuse could result in an unfairly assigned burden of recourse (Sharma, Henderson, and Ghosh 2020), threatening the equality of opportunity (Bell et al. 2024). Also, even if all immutable features are protected, there may exist proxies that are theoretically mutable, but preserve sufficient information about the principals to hinder these protections. Indeed, deciding on the actionability of features remains a major open challenge in the AR literature (Venkatasubramanian and Alfano 2020).

Plausibility is costly. As noted by Altmeyer et al. (2024), more plausible counterfactuals are inevitably more costly. CT improves plausibility and robustness, but it can impact average costs and validity when cheap, implausible and adversarial explanations are removed from the solution space.

CT increases the training times. Just like contrastive and robust learning, CT is more resource-intensive than conventional regimes. Three factors mitigate this effect: (1) CT yields itself to parallel execution; (2) it amortizes the cost of CEs for the training samples; and (3) it can be used to fine-tune conventionally-trained models.

We also highlight three key directions for future research. Firstly, it is an interesting challenge to extend CT beyond classification settings. Our formulation relies on the distinction between non-target class(es) and target class(es), requiring the output space to be discrete. Thus, it does not apply to ML tasks where the change in outcome cannot be readily discretized. Focus on classification is a common choice in research on CEs and AR; other settings have attracted some interest, e.g., regression (Spooner et al. 2021), but there is little consensus how to robustly extend the notion of CEs.

Secondly, our analysis covers CE generators with different characteristics, but it is interesting to extend it to more algorithms, including ones that do not rely on computationally costly gradient-based optimization. This should reduce training costs while possibly preserving the benefits of CT.

Finally, we believe that it is possible to considerably improve hyperparameter selection procedures, and thus performance. We have relied exclusively on grid searches, but future work could benefit from more sophisticated approaches.

6 Conclusion

State-of-the-art machine learning models are prone to learning complex representations that cannot be interpreted by humans. Existing explainability solutions cannot guarantee that explanations agree with these learned representation. As a step towards addressing this challenge, we introduce counterfactual training, a novel training regime that integrates recent advances in contrastive learning, adversarial robustness, and counterfactual explanations to incentivize highly-explainable models. Through extensive experiments, we demonstrate that CT satisfies this goal while preserving the predictive performance and promoting robustness of models. Explanations generated from CT-based models are both more plausible (compliant with the underlying data-generating process) and more actionable (compliant with user-specified mutability constraints), and thus meaningful to their recipients. In turn, our work highlights the value of simultaneously improving models and their explanations.

References

Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. "Counterfactual Vision and Language Learning." In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10041–51. https://doi.org/10.1109/CVPR42600.2020.01006.

Altmeyer, Patrick, Arie van Deursen, and Cynthia C. S. Liem. 2023. "Explaining Black-Box Models through Counterfactuals." In *Proceedings of the JuliaCon Conferences*, 1:130.

Altmeyer, Patrick, Mojtaba Farmanbar, Arie van Deursen, and Cynthia C. S. Liem. 2024. "Faithful Model Explanations through Energy-Constrained Conformal Counterfactuals." In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence*, 38:10829–37. 10. https://doi.org/10.1609/aaai.v38i10.28956.

Augustin, Maximilian, Alexander Meinke, and Matthias Hein. 2020. "Adversarial Robustness on In- and Out-Distribution Improves Explainability." In *Computer Vision – ECCV 2020*, edited by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, 228–45. Cham: Springer.

Balashankar, Ananth, Xuezhi Wang, Yao Qin, Ben Packer, Nithum Thain, Ed Chi, Jilin Chen, and Alex Beutel. 2023. "Improving Classifier Robustness through Active Generative Counterfactual Data Augmentation." In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 127–39. ACL. https://doi.org/10.18653/v1/2023.findings-emnlp.10.

Becker, Barry, and Ronny Kohavi. 1996. "Adult." UCI Machine Learning Repository.

Bell, Andrew, Joao Fonseca, Carlo Abrate, Francesco Bonchi, and Julia Stoyanovich. 2024. "Fairness in Algorithmic Recourse Through the Lens of Substantive Equality of Opportunity." https://arxiv.org/abs/2401.16088.

Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. "Julia: A Fresh Approach to Numerical Computing." *SIAM Review* 59 (1): 65–98. https://doi.org/10.1137/141000671.

Bouchet-Valat, Milan, and Bogumi Kamiski. 2023. "DataFrames.jl: Flexible and Fast Tabular Data in Julia." *Journal of Statistical Software* 107 (4): 1–32. https://doi.org/10.18637/jss.v107.i04.

Byrne, Simon, Lucas C. Wilcox, and Valentin Churavy. 2021. "MPI.jl: Julia Bindings for the Message Passing Interface." *Proceedings of the JuliaCon Conferences* 1 (1): 68. https://doi.org/10.21105/jcon.00068.

Chagas, Ronan Arraes Jardim, Ben Baumgold, Glen Hertz, Hendrik Ranocha, Mark Wells, Nathan Boyer, Nicholas Ritchie, et al. 2024. "Ronisbr/PrettyTables.jl: V2.4.0." Zenodo. https://doi.org/10.5281/zenodo.13835553.

Christ, Simon, Daniel Schwabeneder, Christopher Rackauckas, Michael Krabbe Borregaard, and Thomas Breloff. 2023. "Plots.jl – a User Extendable Plotting API for the Julia Programming Language." https://doi.org/https://doi.org/10.5334/jors.431.

Danisch, Simon, and Julius Krumbiegel. 2021. "Makie.jl: Flexible High-Performance Data Visualization for Julia." *Journal of Open Source Software* 6 (65): 3349. https://doi.org/10.21105/joss.03349.

Du, Yilun, and Igor Mordatch. 2020. "Implicit Generation and Generalization in Energy-Based Models." https://arxiv.org/abs/1903.08689.

Freiesleben, Timo. 2022. "The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples." *Minds and Machines* 32 (1): 77–109.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy. 2015. "Explaining and Harnessing Adversarial Examples." https://arxiv.org/abs/1412.6572.

Grathwohl, Will, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2020. "Your Classifier Is Secretly an Energy Based Model and You Should Treat It Like One." In *International Conference on Learning Representations*.

Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. "A Kernel Two-Sample Test." *The Journal of Machine Learning Research* 13 (1): 723–73.

Guo, Hangzhi, Thanh H. Nguyen, and Amulya Yadav. 2023. "CounterNet: End-to-End Training of Prediction Aware Counterfactual Explanations." In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 577--589. KDD '23. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3580305.3599290.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer New York. https://doi.org/10.1007/978-0-387-84858-7.

Innes, Michael, Elliot Saba, Keno Fischer, Dhairya Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan Karmali, Avik Pal, and Viral Shah. 2018. "Fashionable Modelling with Flux." https://arxiv.org/abs/1811.01457.

Innes, Mike. 2018. "Flux: Elegant Machine Learning with Julia." *Journal of Open Source Software* 3 (25): 602. https://doi.org/10.21105/joss.00602.

Joshi, Shalmali, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. "Towards Realistic Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems." https://arxiv.org/abs/1907.09615.

Kaggle. 2011. "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." https://www.kaggle.com/c/GiveMeSomeCredit; Kaggle. https://www.kaggle.com/c/GiveMeSomeCredit.

Karimi, Amir-Hossein, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. 2021. "A Survey of Algorithmic Recourse: Definitions, Formulations, Solutions, and Prospects." https://arxiv.org/abs/2010.04050.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6405–16. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.

LeCun, Yann. 1998. "The MNIST database of handwritten digits." http://yann.lecun.com/exdb/mnist/.

Lippe, Phillip. 2024. "UvA Deep Learning Tutorials." https://uvadlc-notebooks.readthedocs.io/en/latest/.

Luu, Hoai Linh, and Naoya Inoue. 2023. "Counterfactual Adversarial Training for Improving Robustness of Pretrained Language Models." In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, 881–88. ACL. https://aclanthology.org/2023.paclic-1.88/.

Molnar, Christoph. 2022. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2nd ed. https://christophm.github.io/interpretable-ml-book.

Murphy, Kevin P. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press.

Pace, R Kelley, and Ronald Barry. 1997. "Sparse Spatial Autoregressions." *Statistics & Probability Letters* 33 (3): 291–97. https://doi.org/10.1016/s0167-7152(96)00140-x.

Pawelczyk, Martin, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. "Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis." In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, edited by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, 151:4574–94. Proceedings of Machine Learning Research. PMLR. https://proceedings.mlr.press/v151/pawelczyk22a.html.

Ross, Alexis, Himabindu Lakkaraju, and Osbert Bastani. 2024. "Learning Models for Actionable Recourse." In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS '21. Red Hook, NY, USA: Curran Associates Inc.

Sauer, Axel, and Andreas Geiger. 2021. "Counterfactual Generative Networks." https://arxiv.org/abs/2101.06046.

Schut, Lisa, Oscar Key, Rory McGrath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. 2021. "Generating Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties." In *International Conference on Artificial Intelligence and Statistics*, 1756–64. PMLR.

Sharma, Shubham, Jette Henderson, and Joydeep Ghosh. 2020. "CERTIFAI: A Common Framework to Provide Explanations and Analyse the Fairness and Robustness of Black-box Models." In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 166–72. AIES '20. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3375627.3375812.

Spooner, Thomas, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, and Daniele Magazzeni. 2021. "Counterfactual Explanations for Arbitrary Regression Models." https://arxiv.org/abs/2106.15212.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. 2017. "Axiomatic Attribution for Deep Networks." https://arxiv.org/abs/1703.01365.

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. "Intriguing Properties of Neural Networks." https://arxiv.org/abs/1312.6199.

Teh, Yee Whye, Max Welling, Simon Osindero, and Geoffrey E. Hinton. 2003. "Energy-Based Models for Sparse Overcomplete Representations." *J. Mach. Learn. Res.* 4 (null): 1235–60.

Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. "Learning What Makes a Difference from Counterfactual Examples and Gradient Supervision." In *Computer Vision - ECCV 2020*, 580–99. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-030-58607-2_34.

Ustun, Berk, Alexander Spangher, and Yang Liu. 2019. "Actionable Recourse in Linear Classification." In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 10–19. https://doi.org/10.1145/3287560.3287566.

Venkatasubramanian, Suresh, and Mark Alfano. 2020. "The Philosophical Basis of Algorithmic Recourse." In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 284–93. FAT* '20. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3351095.3372876.

Verma, Sahil, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, and Chirag Shah. 2022. "Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review." https://arxiv.org/abs/2010.10596.

Wachter, Sandra, Brent Mittelstadt, and Chris Russell. 2017. "Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR." *Harv. JL & Tech.* 31: 841. https://doi.org/10.2139/ssrn.3063289.

Wilson, Andrew Gordon. 2020. "The Case for Bayesian Deep Learning." https://arxiv.org/abs/2001.10995.

Wu, Tongshuang, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. "Polyjuice: Generating Counterfactuals for Explaining, Evaluating, and Improving Models." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 6707–23. Online: ACL. https://doi.org/10.18653/v1/2021.acl-long.523.

Yeh, I-Cheng. 2016. "Default of Credit Card Clients." UCI Machine Learning Repository.

Appendix A Notation

Below we provide an overview of some notation used frequently throughout the paper:

- y^+: The target class and also the index of the target class.
- y^-: The non-target class and also the index of non-the target class.
- $\mathbf{x}$: a single training sample.
- $\mathbf{x}'$: a counterfactual.
- $\mathbf{x}^+$: a training sample in the target class (ground-truth).
- $\mathbf{y}^+$: The one-hot encoded output vector for the target class.
- θ: Model parameters (unspecified).
- Θ: Matrix of parameters.
- $\mathbf{M}(\cdot)$: linear predictions (logits) of the classifier.

A.1 Other Technical Details

Maximum mean discrepancy is defined as follows,

$$\begin{aligned}\text{MMD}(X', \tilde{X}') = &\frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(\tilde{x}_i, \tilde{x}_j) \\ &- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, \tilde{x}_j)\end{aligned} \tag{7}$$

where $k(\cdot, \cdot)$ is a kernel function (Gretton et al. 2012). We make use of a Gaussian kernel with a constant length-scale parameter of 0.5. In our implementation, Equation 7 is by default applied to the entire subset of the training data for which $y = y^+$.

Appendix B Technical Details of Our Approach

B.1 Generating Counterfactuals through Gradient Descent

In this section, we provide some background on gradient-based counterfactual generators (Section B.1.1) and discuss how we define convergence in this context (Section B.1.2).

B.1.1 Background

Gradient-based counterfactual search was originally proposed by Wachter, Mittelstadt, and Russell (2017). It generally solves the following unconstrained objective,

$$\min_{\mathbf{z}' \in \mathcal{Z}^L} \{\text{yloss}(\mathbf{M}_\theta(g(\mathbf{z}')), \mathbf{y}^+) + \lambda \text{cost}(g(\mathbf{z}'))\}$$

where $g : \mathcal{Z} \mapsto \mathcal{X}$ is an invertible function that maps from the L-dimensional counterfactual state space to the feature space and $\text{cost}(\cdot)$ denotes one or more penalties that are used to induce certain properties of the counterfactual outcome. As above, $\mathbf{y}^+$ denotes the target output and $\mathbf{M}_\theta(\mathbf{x})$ returns the logit predictions of the underlying classifier for $\mathbf{x} = g(\mathbf{z})$.

For all generators used in this work we use standard logit crossentropy loss for $\text{yloss}(\cdot)$. All generators also penalize the distance (ℓ_1-norm) of counterfactuals from their original factual state. For *Generic* and *ECCo*, we have $\mathcal{Z} \coloneqq \mathcal{X}$ and $g(\mathbf{z}) = g(\mathbf{z})^{-1} = \mathbf{z}$, that is counterfactual are searched directly in the feature space. Conversely, *REVISE* traverses the latent space of a variational autoencoder (VAE) fitted to the training data, where $g(\cdot)$ corresponds to the decoder (Joshi et al. 2019). In addition to the distance penalty, *ECCo* uses an additional penalty component that regularizes the energy associated with the counterfactual, $\mathbf{x}'$ (Altmeyer et al. 2024).

B.1.2 Convergence

An important consideration when generating counterfactual explanations using gradient-based methods is how to define convergence. Two common choices are to 1) perform gradient descent over a fixed number of iterations T, or 2) conclude the search as soon as the predicted probability for the target class has reached a pre-determined threshold, τ: $\mathcal{S}(\mathbf{M}_\theta(\mathbf{x}'))[y^+] \geq \tau$. We prefer the latter for our purposes, because it explicitly defines convergence in terms of the black-box model, $\mathbf{M}(\mathbf{x})$.

Defining convergence in this way allows for a more intuitive interpretation of the resulting counterfactual outcomes than with fixed T. Specifically, it allows us to think of counterfactuals as explaining 'high-confidence' predictions by the model for the target class y^+. Depending on the context and application, different choices of τ can be considered as representing 'high-confidence' predictions.

B.2 Protecting Mutability Constraints with Linear Classifiers

In Section 3.4 we explain that to avoid penalizing implausibility that arises due to mutability constraints, we impose a point mass prior on $p(\mathbf{x})$ for the corresponding feature. We argue in Section 3.4 that this approach induces models to be less sensitive to immutable features and demonstrate this empirically in Section 4. Below we derive the analytical results in Prp.~3.1.

Proof. Let d_{mtbl} and d_{immtbl} denote some mutable and immutable feature, respectively. Suppose that $\mu_{y^-,d_{\text{immtbl}}} < \mu_{y^+,d_{\text{immtbl}}}$ and $\mu_{y^-,d_{\text{mtbl}}} > \mu_{y^+,d_{\text{mtbl}}}$, where $\mu_{k,d}$ denotes the conditional sample mean of feature d in class k. In words, we assume that the immutable feature tends to take lower values for samples in the non-target class y^- than in the target class y^+. We assume the opposite to hold for the mutable feature.

Assuming multivariate Gaussian class densities with common diagonal covariance matrix $\Sigma_k = \Sigma$ for all $k \in \mathcal{K}$, we have for the log likelihood ratio between any two classes $k, m \in \mathcal{K}$ (Hastie, Tibshirani, and Friedman 2009):

$$\log \frac{p(k|\mathbf{x})}{p(m|\mathbf{x})} = \mathbf{x}^\top \Sigma^{-1}(\mu_k - \mu_m) + \text{const} \tag{8}$$

By independence of $x_1, ..., x_D$, the full log-likelihood ratio decomposes into:

$$\log \frac{p(k|\mathbf{x})}{p(m|\mathbf{x})} = \sum_{d=1}^{D} \frac{\mu_{k,d} - \mu_{m,d}}{\sigma_d^2} x_d + \text{const} \tag{9}$$

By the properties of our classifier (*multinomial logistic regression*), we have:

$$\log \frac{p(k|\mathbf{x})}{p(m|\mathbf{x})} = \sum_{d=1}^{D} \left(\theta_{k,d} - \theta_{m,d}\right) x_d + \text{const} \tag{10}$$

where $\theta_{k,d} = \Theta[k, d]$ denotes the coefficient on feature d for class k.

Based on Equation 9 and Equation 10 we can identify that $(\mu_{k,d} - \mu_{m,d}) \propto (\theta_{k,d} - \theta_{m,d})$ under the assumptions we made above. Hence, we have that $(\theta_{y^-,d_{\text{immtbl}}} - \theta_{y^+,d_{\text{immtbl}}}) < 0$ and $(\theta_{y^-,d_{\text{mtbl}}} - \theta_{y^+,d_{\text{mtbl}}}) > 0$

Let $\mathbf{x}'$ denote some randomly chosen individual from class y^- and let $y^+ \sim p(y)$ denote the randomly chosen target class. Then the partial derivative of the contrastive divergence penalty Equation 3 with respect to coefficient $\theta_{y^+,d}$ is equal to

$$\frac{\partial}{\partial \theta_{y^+,d}} \left(\text{div}(\mathbf{x}^+, \mathbf{x}', \mathbf{y}; \theta)\right) = \frac{\partial}{\partial \theta_{y^+,d}} \left((-\mathbf{M}_\theta(\mathbf{x}^+)[y^+]) - (-\mathbf{M}_\theta(\mathbf{x}')[y^+])\right) = x'_d - x^+_d \tag{11}$$

and equal to zero everywhere else.

Since $(\mu_{y^-,d_{\text{immtbl}}} < \mu_{y^+,d_{\text{immtbl}}})$ we are more likely to have $(x'_{d_{\text{immtbl}}} - x^+_{d_{\text{immtbl}}}) < 0$ than vice versa at initialization. Similarly, we are more likely to have $(x'_{d_{\text{mtbl}}} - x^+_{d_{\text{mtbl}}}) > 0$ since $(\mu_{y^-,d_{\text{mtbl}}} > \mu_{y^+,d_{\text{mtbl}}})$.

This implies that if we do not protect feature d_{immtbl}, the contrastive divergence penalty will decrease $\theta_{y^-,d_{\text{immtbl}}}$ thereby exacerbating the existing effect $(\theta_{y^-,d_{\text{immtbl}}} - \theta_{y^+,d_{\text{immtbl}}}) < 0$. In words, not protecting the immutable feature would have the undesirable effect of making the classifier more sensitive to this feature, in that it would be more likely to predict class y^- as opposed to y^+ for lower values of d_{immtbl}.

By the same rationale, the contrastive divergence penalty can generally be expected to increase $\theta_{y^-,d_{\text{mtbl}}}$ exacerbating $(\theta_{y^-,d_{\text{mtbl}}} - \theta_{y^+,d_{\text{mtbl}}}) > 0$. In words, this has the effect of making the classifier more sensitive to the mutable feature, in that it would be more likely to predict class y^- as opposed to y^+ for higher values of d_{mtbl}.

Thus, our proposed approach of protecting feature d_{immtbl} has the net affect of decreasing the classifier's sensitivity to the immutable feature relative to the mutable feature (i.e. no change in sensitivity for d_{immtbl} relative to increased sensitivity for d_{mtbl}). □

B.3 Domain Constraints

We apply domain constraints on counterfactuals during training and evaluation. There are at least two good reasons for doing so. Firstly, within the context of explainability and algorithmic recourse, real-world attributes are often domain constrained: the *age* feature, for example, is lower bounded by zero and upper bounded by the maximum human lifespan. Secondly, domain constraints help mitigate training instabilities commonly associated with energy-based modelling (Grathwohl et al. 2020; Altmeyer et al. 2024).

For our image datasets, features are pixel values and hence the domain is constrained by the lower and upper bound of values that pixels can take depending on how they are scaled (in our case $[-1, 1]$). For all other features d in our synthetic and tabular datasets, we automatically infer domain constraints $[x_d^{\text{LB}}, x_d^{\text{UB}}]$ as follows,

$$\begin{aligned} x_d^{\text{LB}} &= \arg\min_{x_d}\{\mu_d - n_{\sigma_d}\sigma_d, \arg\min_{x_d} x_d\} \\ x_d^{\text{UB}} &= \arg\max_{x_d}\{\mu_d + n_{\sigma_d}\sigma_d, \arg\max_{x_d} x_d\} \end{aligned} \tag{12}$$

where μ_d and σ_d denote the sample mean and standard deviation of feature d. We set $n_{\sigma_d} = 3$ across the board but higher values and hence wider bounds may be appropriate depending on the application.

B.4 Training Hyperparameters

Note 1 presents the default hyperparameters used during training.

Note 1: Training Phase

- Meta Parameters:
  - Generator: `ecco`
  - Model: `mlp`
- Model:
  - Activation: `relu`
  - No. Hidden: `32`
  - No. Layers: `1`
- Training Parameters:
  - Burnin: `0.0`
  - Class Loss: `logitcrossentropy`
  - Convergence: `threshold`
  - Generator Parameters:
    - Decision Threshold: `0.75`
    - λ_{cst}: `0.001`
    - λ_{egy}: `5.0`
    - Learning Rate: `0.25`
    - Maximum Iterations: `30`
    - Optimizer: `sgd`
    - Type: `ECCo`
  - λ_{adv}: `0.25`
  - λ_{clf}: `1.0`
  - λ_{div}: `0.5`
  - λ_{reg}: `0.1`
  - Learning Rate: `0.001`
  - No. Counterfactuals: `1000`
  - No. Epochs: `100`
  - Objective: `full`
  - Optimizer: `adam`

B.5 Evaluation Details

For all of our evaluations, we proceed as follows: for each experiment setting we generate multiple counterfactuals ("No. Counterfactuals"), randomly choosing the factual and target class each time (Note 2). We do this across multiple rounds ("No. Runs") with different random seeds to account for stochasticity (Note 2). This is in line with standard practice in the related literature on CE. Note 2 presents the default hyperparameters used during evaluation. For our final results presented in the main paper, we rely on held out test sets to sample factuals (and outputs for our performance metrics). For tuning purposes we rely on training or validation sets.

B.5.1 Robust Accuracy

To evaluate robust accuracy (Acc.*), we use the Fast Gradient Sign Method (FGSM) to perturb test samples (Goodfellow, Shlens, and Szegedy 2015). For the main results, we have set the perturbation size to $\epsilon = 0.03$. We have also tested other perturbation sizes, as well as randomly perturbed data. Although not reported here, we have consistently found strong outperformance of CT compared to the weak baseline.

Note 2: Evaluation Phase

- Counterfactual Parameters:
  - Convergence: `threshold`
  - Decision Threshold: `0.95`
  - Generator Parameters:
    - Decision Threshold: `0.75`
    - λ_{cst}: `0.001`
    - λ_{egy}: `5.0`
    - Learning Rate: `0.25`
    - Maximum Iterations: `30`
    - Optimizer: `sgd`
    - Type: `ECCo`
  - Maximum Iterations: `50`
  - No. Individuals: `100`
  - No. Runs: `5`

Appendix C Details on Main Experiments

C.1 Final Hyperparameters

As discussed Section 4, CT is sensitive to certain hyperparameter choices. We study the effect of many hyperparameters extensively in Section D. For the main results, we tune a small set of key hyperparameters (Section E). The final choices for the main results are presented for each data set in Table 2 along with training, test and batch sizes.

C.2 Final Results

Plus/minus two standard deviations of bootstrap estimates.

Table 2: Final hyperparameters used for the main results presented in Section 4. Any hyperparameter not shown here is set to its default value (Note 1).

Data	No. Train	No. Test	Batchsize	Domain	Decision Threshold	No. Counterfactuals	λ_{reg}
LS	3600	600	30	none	0.5	1000	0.01
Circ	3600	600	30	none	0.5	1000	0.5
Moon	3600	600	30	none	0.9	1000	0.25
OL	3600	600	30	none	0.5	1000	0.25
Adult	26049	5010	1000	none	0.75	5000	0.25
CH	16504	3101	1000	none	0.5	5000	0.25
Cred	10617	1923	1000	none	0.5	5000	0.25
GMSC	13371	2474	1000	none	0.5	5000	0.5
MNIST	11000	2000	1000	(-1.0, 1.0)	0.5	5000	0.01

Table 3: Mean outcomes for **CT** and **BL** along with bootstrapped confidence intervals (99%) for difference in mean outcomes grouped by dataset and evaluation metric. Column **LB** and **UB** show the lower and upper bound of the intervals, respectively, and computed using the percentile method. The underlying counterfactual evaluations are the same as the ones used to produce Table 1.

Variable	**Data**	**CT**	**BL**	**LB**	**UB**
Cost	Adult	2.26	2.2	-0.22	0.28
Cost	CH	1.46	2.46	-1.1	-0.89
Cost	Circ	0.67	1.23	-0.58	-0.53
Cost	Cred	2.68	2.29	0.16	0.63
Cost	GMSC	1.14	3.05	-2.45	-1.77
Cost	LS	3.82	4.44	-0.7	-0.56
Cost	MNIST	77.04	68.67	-3.47	18.34
Cost	Moon	1.55	1.6	-0.08	-0.01
Cost	OL	1.62	2.63	-1.15	-0.81
IP*	Adult	0.07	0.11	-0.06	-0.01
IP*	CH	0.02	0.06	-0.06	-0.04
IP*	Circ	0.0	0.0	-0.01	-0.0
IP*	Cred	0.03	0.06	-0.05	-0.01
IP*	GMSC	0.02	0.07	-0.06	-0.04
IP*	LS	0.1	0.23	-0.14	-0.12
IP*	MNIST	0.04	0.04	-0.1	0.09
IP*	Moon	0.02	0.02	-0.01	-0.0
IP*	OL	0.12	0.09	-0.01	0.05
IP	Adult	15.03	15.15	-0.68	0.26
IP	CH	6.61	7.52	-1.17	-0.63
IP	Circ	1.03	2.36	-1.37	-1.29
IP	Cred	19.31	22.03	-3.69	-1.74
IP	GMSC	6.19	8.09	-2.4	-1.49
IP	LS	2.41	3.4	-1.04	-0.94
IP	MNIST	258.83	278.54	-30.49	-7.64
IP	Moon	1.36	1.71	-0.38	-0.32
IP	OL	4.49	4.44	-0.03	0.13

Table 4: Costs

Data	**Cost** $(-\%)$
LS	-26.82±0.86 *
Circ	40.97±0.82 *
Moon	33.83±0.98 *
OL	10.35±1.28 *
Adult	1.16±3.53
CH	-34.89±2.31 *
Cred	28.24±1.08 *
GMSC	3.54±5.78
MNIST	-31.67±7.72 *
Avg.	2.75

C.2.1 Robust Performance Plots

C.2.2 Confidence Intervals

C.2.3 Qualitative Findings for Image Data

Figure 4 shows much more plausible (faithful) counterfactuals for a model with CT than the model with conventional training (Figure 5).

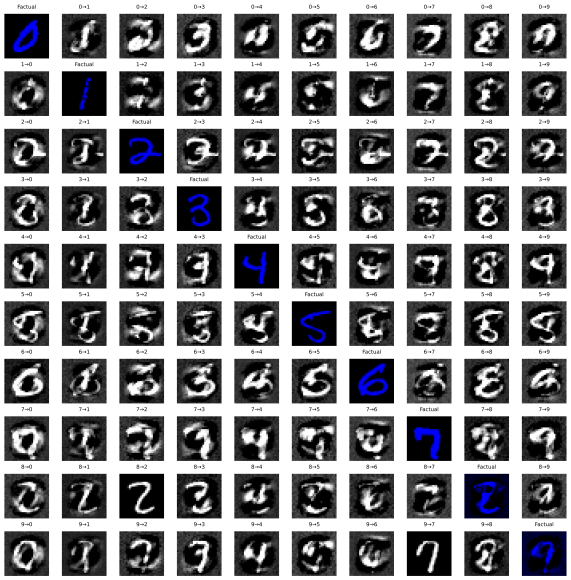

Figure 4: Counterfactual images for *MLP* with counterfactual training. Factual images are shown on the diagonal, with the corresponding counterfactual for each target class (columns) in that same row. The underlying generator, *ECCo*, aims to generate counterfactuals that are faithful to the model (Altmeyer et al. 2024).

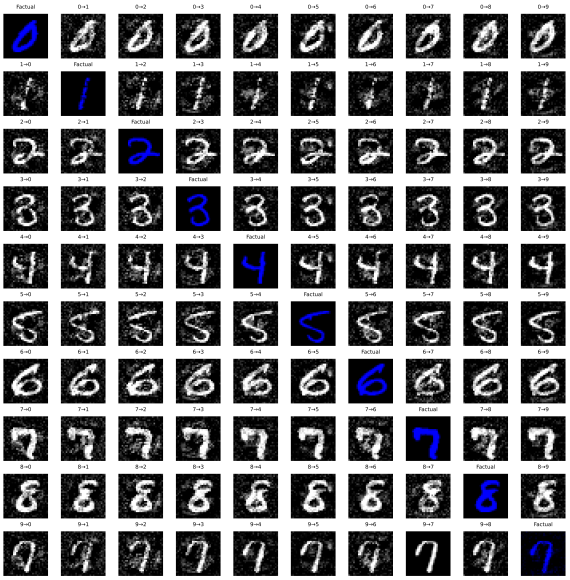

Figure 5: The same setup, factuals, model architecture and generator as in Figure 4, but the model was trained conventionally.

Table 5: Validity

Data	CT	BL
LS	1.0	1.0
Circ	0.97	0.52
Moon	1.0	1.0
OL	0.87	0.98
Adult	0.61	0.99
CH	0.96	1.0
Cred	0.7	1.0
GMSC	0.63	1.0
MNIST	1.0	1.0

C.2.4 Costs

C.2.5 Validity

Table 6: Validity

Data	CT	BL
LS	1.0	1.0
Circ	0.67	0.49
Moon	0.99	0.98
OL	0.37	0.57
Adult	0.56	0.99
CH	0.96	1.0
Cred	0.67	1.0
GMSC	0.38	1.0
MNIST	1.0	1.0

Appendix D Grid Searches

To assess the hyperparameter sensitivity of our proposed training regime we ran multiple large grid searches for all of our synthetic datasets. We have grouped these grid searches into multiple categories:

1. **Generator Parameters** (Section D.2): Investigates the effect of changing hyperparameters that affect the counterfactual outcomes during the training phase.
2. **Penalty Strengths** (Section D.3): Investigates the effect of changing the penalty strengths in out proposed objective (Equation 2).
3. **Other Parameters** (Section D.4): Investigates the effect of changing other training parameters, including the total number of generated counterfactuals in each epoch.

We begin by summarizing the high-level findings in Section D.1.2. For each of the categories, Section D.2 to Section D.4 then present all details including the exact parameter grids, average predictive performance outcomes and key evaluation metrics for the generated counterfactuals.

D.1 Evaluation Details

To measure predictive performance, we compute the accuracy and F1-score for all models on test data (Table 7, Table 8, Table 9). With respect to explanatory performance, we report here our findings for the (im)plausibility and cost of counterfactuals at test time. Since the computation of our proposed divergence metric (Equation 6) is memory-intensive, we rely on the distance-based metric for the grid searches. For the counterfactual evaluation, we draw factual samples from the training data for the grid searches to avoid data leakage with respect to our final results reported in the body of the paper. Specifically, we want to avoid choosing our default hyperparameters based on results on the test data. Since we are optimizing for explainability, not predictive performance, we still present test accuracy and F1-scores.

D.1.1 Predictive Performance

We find that CT is associated with little to no decrease in average predictive performance for our synthetic datasets: test accuracy and F1-scores decrease by at most ~1 percentage point, but generally much less (Table 7, Table 8, Table 9). Variation across hyperparameters is negligible as indicated by small standard deviations for these metrics across the board.

D.1.2 Counterfactual Outcomes

Overall, we find that counterfactual training achieves it key objectives consistently across all hyperparameter settings and also broadly across datasets: plausibility is improved by up to 60 percent (%) for the *Circles* data (e.g. Figure 6), 25-30% for the *Moons* data (e.g. Figure 8) and 10-20% for the *Linearly Separable* data (e.g. Figure 7). At the same time, the average costs of faithful counterfactuals are reduced in many cases by around 20-25% for *Circles* (e.g. Figure 10) and up to 50% for *Moons* (e.g. Figure 12). For the *Linearly Separable* data, costs are generally increased although typically by less than 10% (e.g. Figure 11), which reflects a common tradeoff between costs and plausibility (Altmeyer et al. 2024).

We do observe strong sensitivity to certain hyperparameters, with clear an manageable patterns. Concerning generator parameters, we firstly find that using *REVISE* to generate counterfactuals during training typically yields the worst outcomes out of all generators, often leading to a substantial decrease in plausibility. This finding can be attributed to the fact that *REVISE* effectively assigns the task of learning plausible explanations from the model itself to a surrogate VAE. In other words, counterfactuals generated by *REVISE* are less faithful to the model that *ECCo* and *Generic*, and

hence we would expect them to be a less effective and, in fact, potentially detrimental role in our training regime. Secondly, we observe that allowing for a higher number of maximum steps T for the counterfactual search generally yields better outcomes. This is intuitive, because it allows more counterfactuals to reach maturity in any given iteration. Looking in particular at the results for *Linearly Separable*, it seems that higher values for T in combination with higher decision thresholds (τ) yields the best results when using *ECCo*. But depending on the degree of class separability of the underlying data, a high decision-threshold can also affect results adversely, as evident from the results for the *Overlapping* data (Figure 9): here we find that CT generally fails to achieve its objective because only a tiny proportion of counterfactuals ever reaches maturity.

Regarding penalty strengths, we find that the strength of the energy regularization, λ_{reg} is a key hyperparameter, while sensitivity with respect to λ_{div} and λ_{adv} is much less evident. In particular, we observe that not regularizing energy enough or at all typically leads to poor performance in terms of decreased plausibility and increased costs, in particular for *Circles* (Figure 14), *Linearly Separable* (Figure 15) and *Overlapping* (Figure 17). High values of λ_{reg} can increase the variability in outcomes, in particular when combined with high values for λ_{div} and λ_{adv}, but this effect is less pronounced.

Finally, concerning other hyperparameters we observe that the effectiveness and stability of CT is positively associated with the number of counterfactuals generated during each training epoch, in particular for *Circles* (Figure 22) and *Moons* (Figure 24). We further find that a higher number of training epochs is beneficial as expected, where we tested training models for 50 and 100 epochs. Interestingly, we find that it is not necessary to employ CT during the entire training phase to achieve the desired improvements in explainability: specifically, we have tested training models conventionally during the first half of training before switching to CT after this initial burn-in period.

D.2 Generator Parameters

The hyperparameter grid with varying generator parameters during training is shown in Note 3. The corresponding evaluation grid used for these experiments is shown in Note 4.

Note 3: Training Phase

- Generator Parameters:
  - Decision Threshold: `0.75, 0.9, 0.95`
  - λ_{egy}: `0.1, 0.5, 5.0, 10.0, 20.0`
  - Maximum Iterations: `5, 25, 50`
- Generator: `ecco, generic, revise`
- Model: `mlp`
- Training Parameters:
  - Objective: `full, vanilla`

Note 4: Evaluation Phase

- Generator Parameters:
  - λ_{egy}: `0.1, 0.5, 1.0, 5.0, 10.0`

D.2.1 Predictive Performance

Predictive performance measures for this grid search are shown in Table 7.

Table 7: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 3) and evaluation-phase parameters (Note 4).

Dataset	Variable	Objective	Mean	Se
Circ	Accuracy	Full	1.0	0.0
Circ	Accuracy	Vanilla	1.0	0.0
Circ	F1-score	Full	1.0	0.0
Circ	F1-score	Vanilla	1.0	0.0
LS	Accuracy	Full	1.0	0.0
LS	Accuracy	Vanilla	1.0	0.0

Continuing table below.

Dataset	Variable	Objective	Mean	Se
LS	F1-score	Full	1.0	0.0
LS	F1-score	Vanilla	1.0	0.0
Moon	Accuracy	Full	1.0	0.0
Moon	Accuracy	Vanilla	1.0	0.0
Moon	F1-score	Full	1.0	0.0
Moon	F1-score	Vanilla	1.0	0.0
OL	Accuracy	Full	0.91	0.0
OL	Accuracy	Vanilla	0.92	0.0
OL	F1-score	Full	0.91	0.0
OL	F1-score	Vanilla	0.92	0.0

D.2.2 Plausibility

The results with respect to the plausibility measure are shown in Figure 6 to Figure 9.

D.2.3 Cost

The results with respect to the cost measure are shown in Figure 10 to Figure 13.

D.3 Penalty Strengths

The hyperparameter grid with varying penalty strengths during training is shown in Note 5. The corresponding evaluation grid used for these experiments is shown in Note 6.

Note 5: Training Phase

- Generator: `ecco, generic, revise`
- Model: `mlp`
- Training Parameters:
  - λ_{adv}: `0.1, 0.25, 1.0`
  - λ_{div}: `0.01, 0.1, 1.0`
  - λ_{reg}: `0.0, 0.01, 0.1, 0.25, 0.5`
  - Objective: `full, vanilla`

Note 6: Evaluation Phase

- Generator Parameters:
  - λ_{egy}: `0.1, 0.5, 1.0, 5.0, 10.0`

D.3.1 Predictive Performance

Predictive performance measures for this grid search are shown in Table 8.

Table 8: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 5) and evaluation-phase parameters (Note 6).

Dataset	Variable	Objective	Mean	Se
Circ	Accuracy	Full	0.99	0.01
Circ	Accuracy	Vanilla	1.0	0.0
Circ	F1-score	Full	0.99	0.01
Circ	F1-score	Vanilla	1.0	0.0
LS	Accuracy	Full	1.0	0.01
LS	Accuracy	Vanilla	1.0	0.0
LS	F1-score	Full	1.0	0.01
LS	F1-score	Vanilla	1.0	0.0
Moon	Accuracy	Full	0.99	0.04

Continuing table below.

Dataset	Variable	Objective	Mean	Se
Moon	Accuracy	Vanilla	1.0	0.01
Moon	F1-score	Full	0.99	0.04
Moon	F1-score	Vanilla	1.0	0.01
OL	Accuracy	Full	0.91	0.02
OL	Accuracy	Vanilla	0.92	0.0
OL	F1-score	Full	0.91	0.02
OL	F1-score	Vanilla	0.92	0.0

D.3.2 Plausibility

The results with respect to the plausibility measure are shown in Figure 14 to Figure 17.

D.3.3 Cost

The results with respect to the cost measure are shown in Figure 18 to Figure 21.

D.4 Other Parameters

The hyperparameter grid with other varying training parameters is shown in Note 7. The corresponding evaluation grid used for these experiments is shown in Note 8.

Note 7: Training Phase

- Generator: `ecco, generic, revise`
- Model: `mlp`
- Training Parameters:
  - Burnin: `0.0, 0.5`
  - No. Counterfactuals: `100, 1000`
  - No. Epochs: `50, 100`
  - Objective: `full, vanilla`

Note 8: Evaluation Phase

- Generator Parameters:
  - λ_{egy}: `0.1, 0.5, 1.0, 5.0, 10.0`

D.4.1 Predictive Performance

Predictive performance measures for this grid search are shown in Table 9.

Table 9: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 7) and evaluation-phase parameters (Note 8).

Dataset	Variable	Objective	Mean	Se
Circ	Accuracy	Full	0.99	0.0
Circ	Accuracy	Vanilla	1.0	0.0
Circ	F1-score	Full	0.99	0.0
Circ	F1-score	Vanilla	1.0	0.0
LS	Accuracy	Full	1.0	0.0
LS	Accuracy	Vanilla	1.0	0.0
LS	F1-score	Full	1.0	0.0
LS	F1-score	Vanilla	1.0	0.0
Moon	Accuracy	Full	1.0	0.01
Moon	Accuracy	Vanilla	0.99	0.02
Moon	F1-score	Full	1.0	0.01
Moon	F1-score	Vanilla	0.99	0.02

Continuing table below.

Dataset	Variable	Objective	Mean	Se
OL	Accuracy	Full	0.91	0.01
OL	Accuracy	Vanilla	0.92	0.0
OL	F1-score	Full	0.91	0.01
OL	F1-score	Vanilla	0.92	0.0

D.4.2 Plausibility

The results with respect to the plausibility measure are shown in Figure 22 to Figure 25.

D.4.3 Cost

The results with respect to the cost measure are shown in Figure 26 to Figure 29.

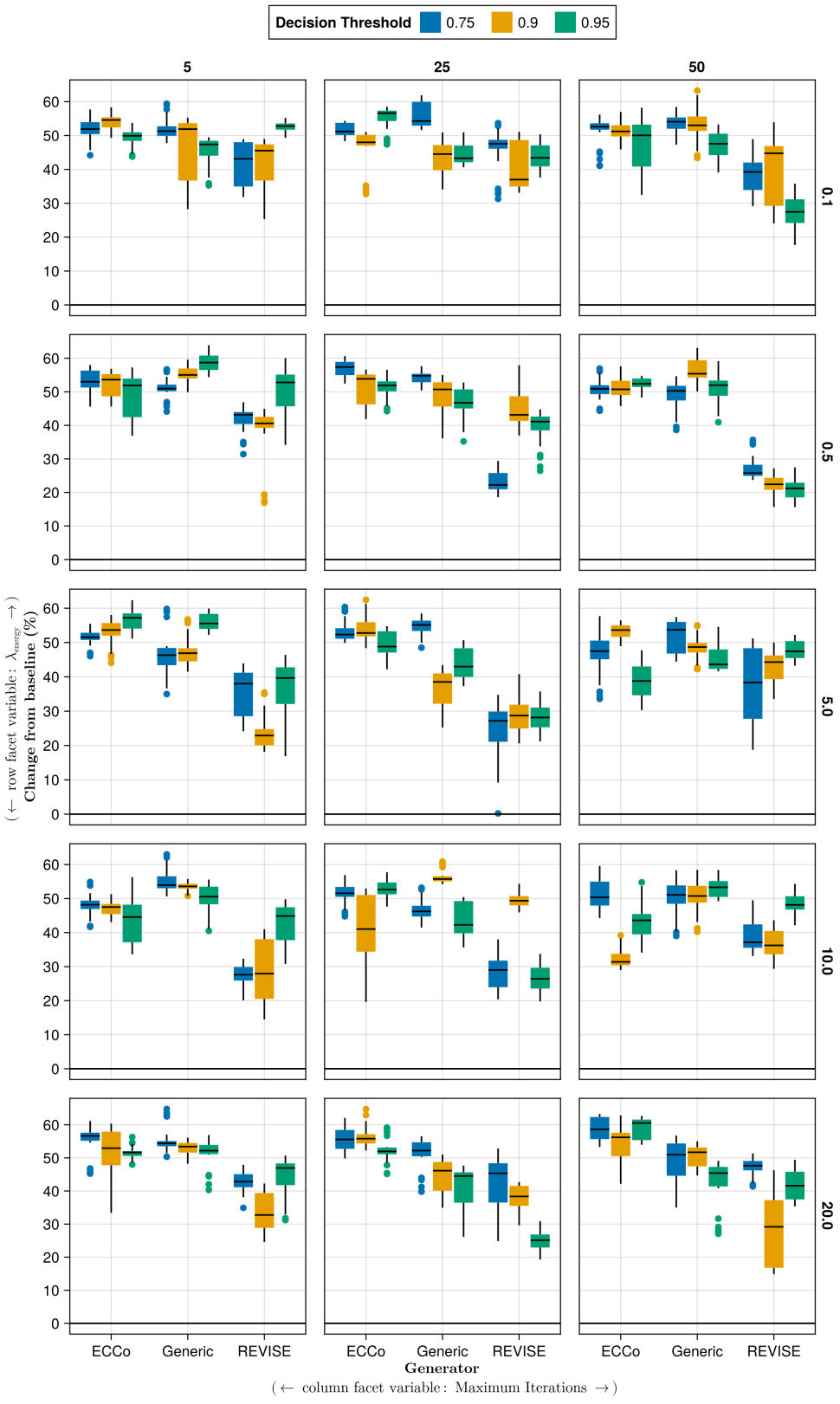

Figure 6: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

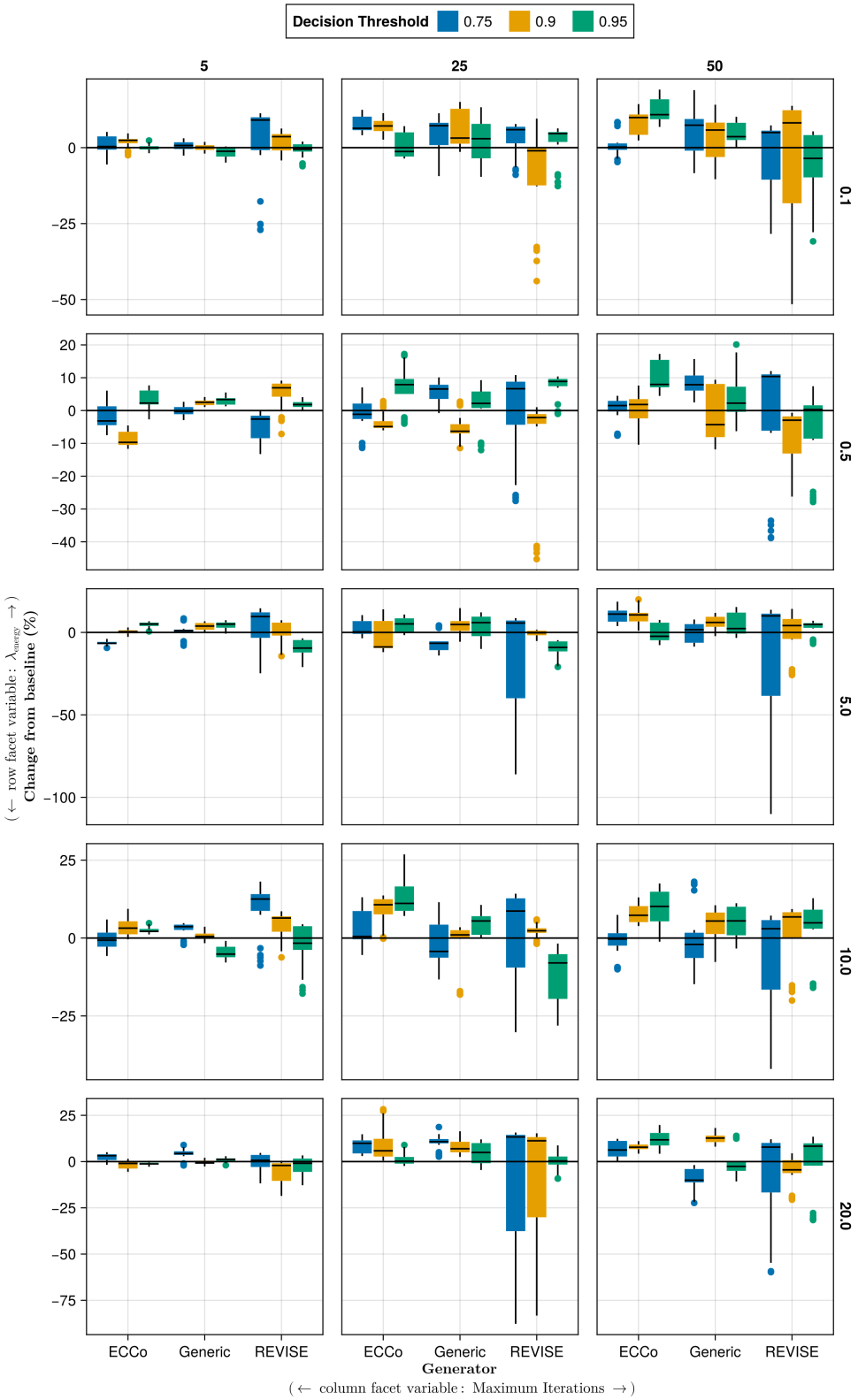

Figure 7: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

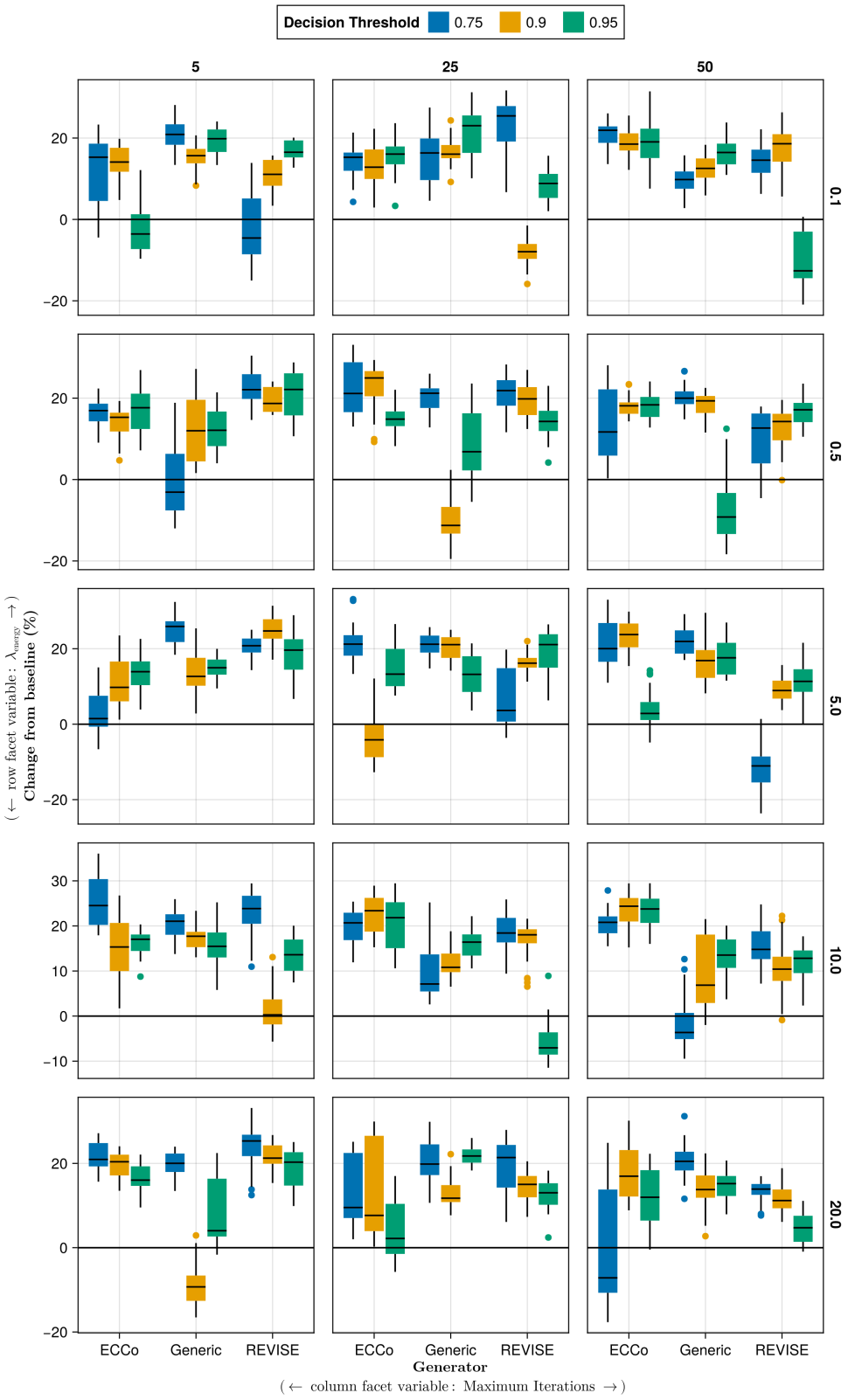

Figure 8: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

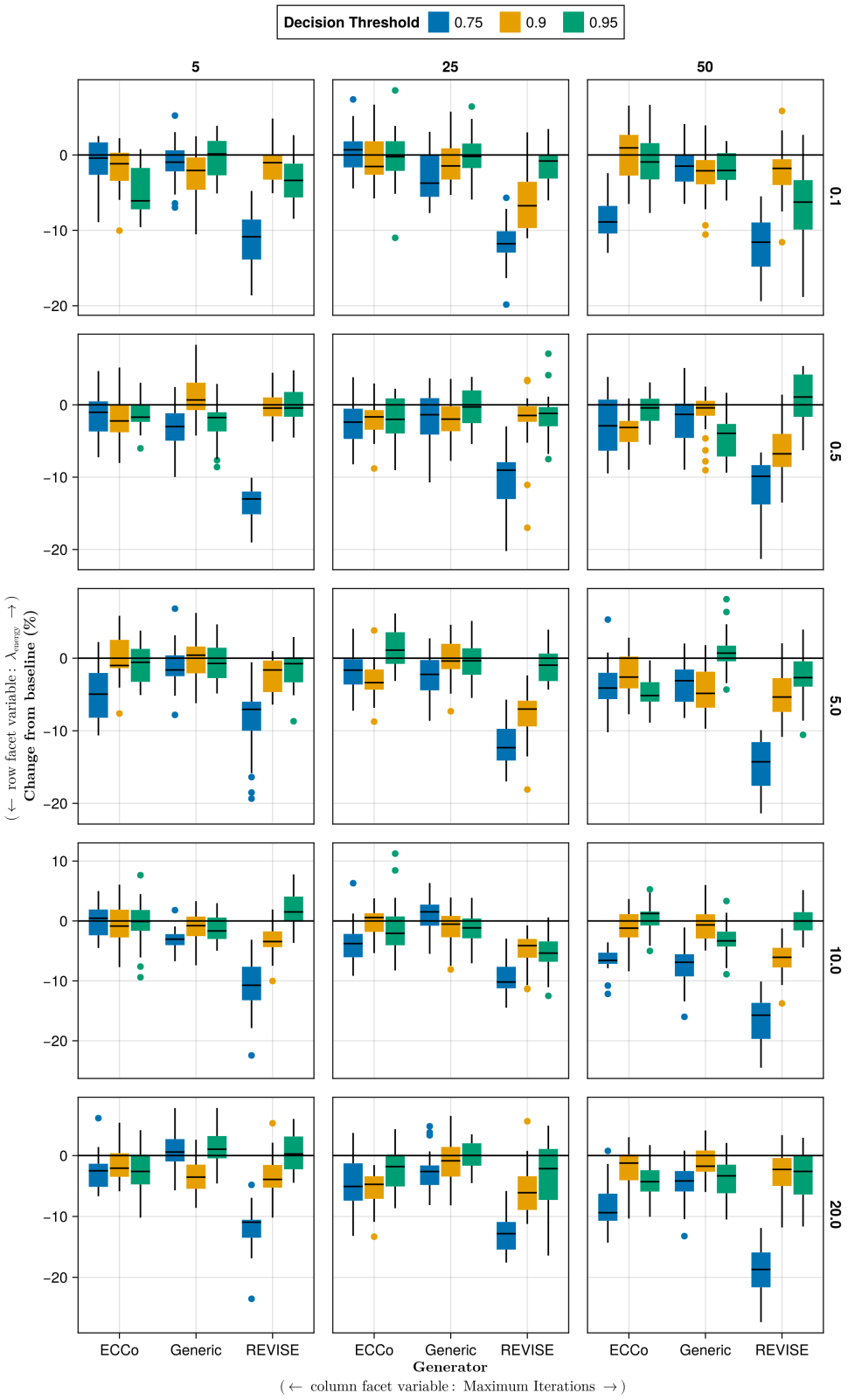

Figure 9: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

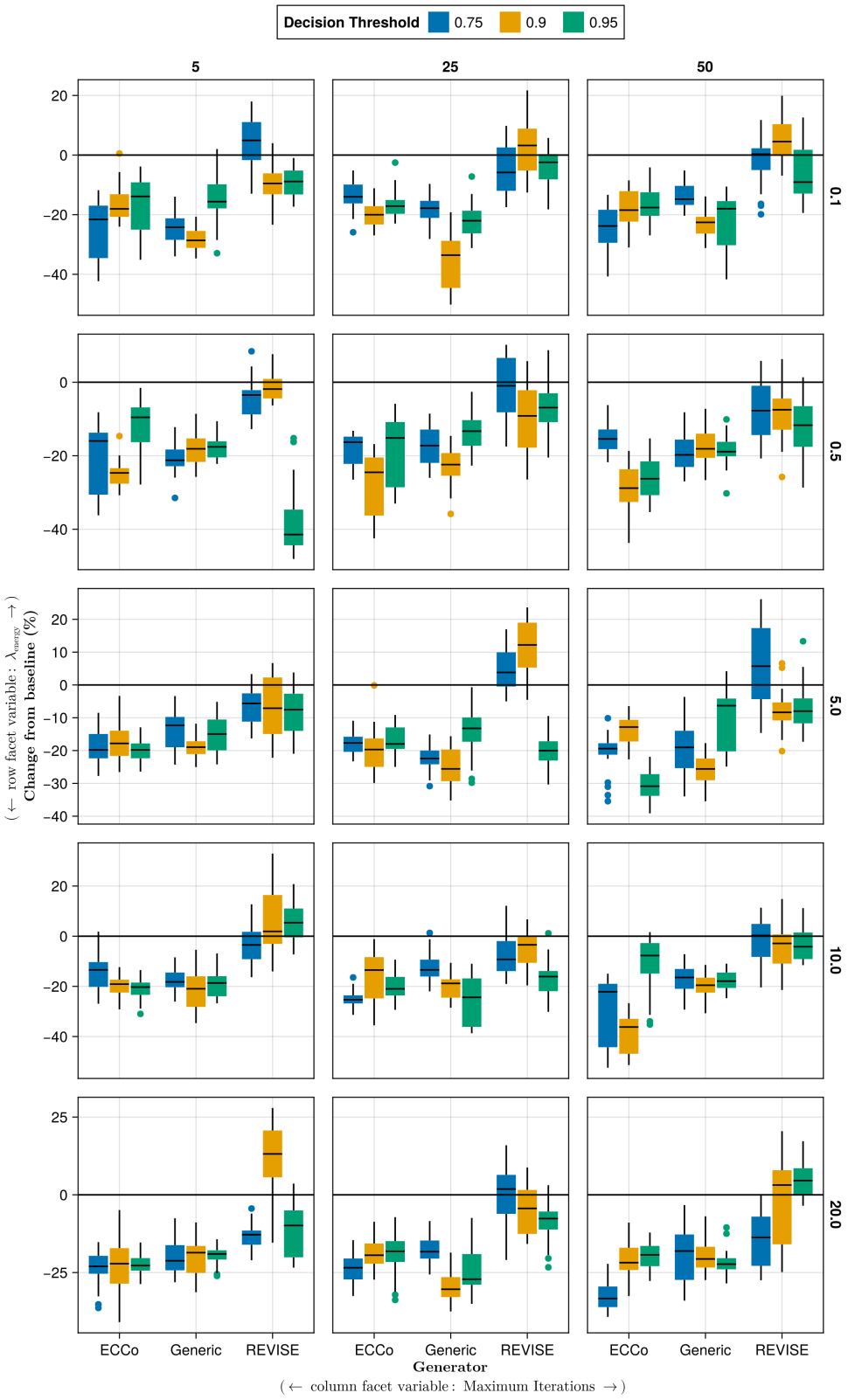

Figure 10: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

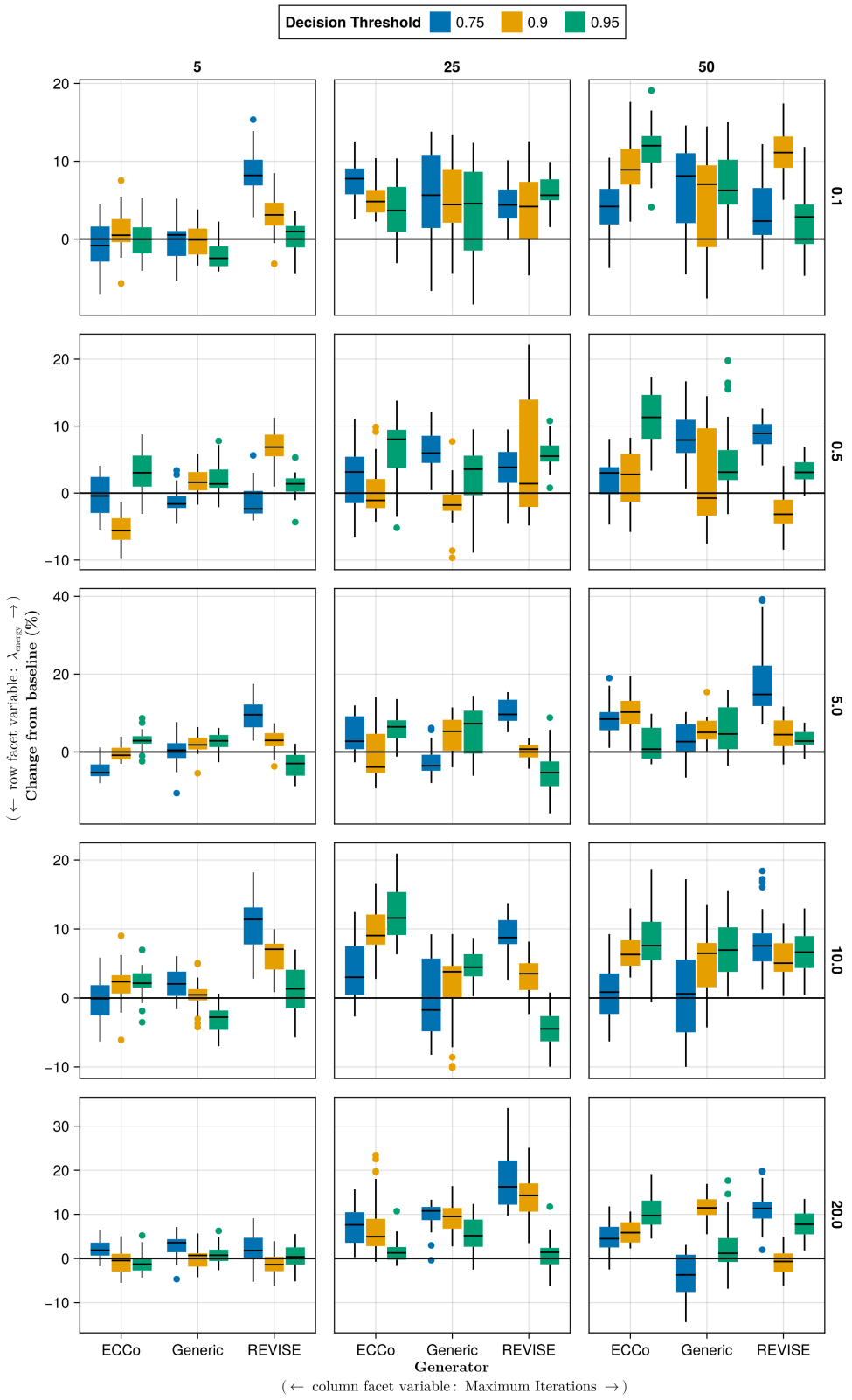

Figure 11: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

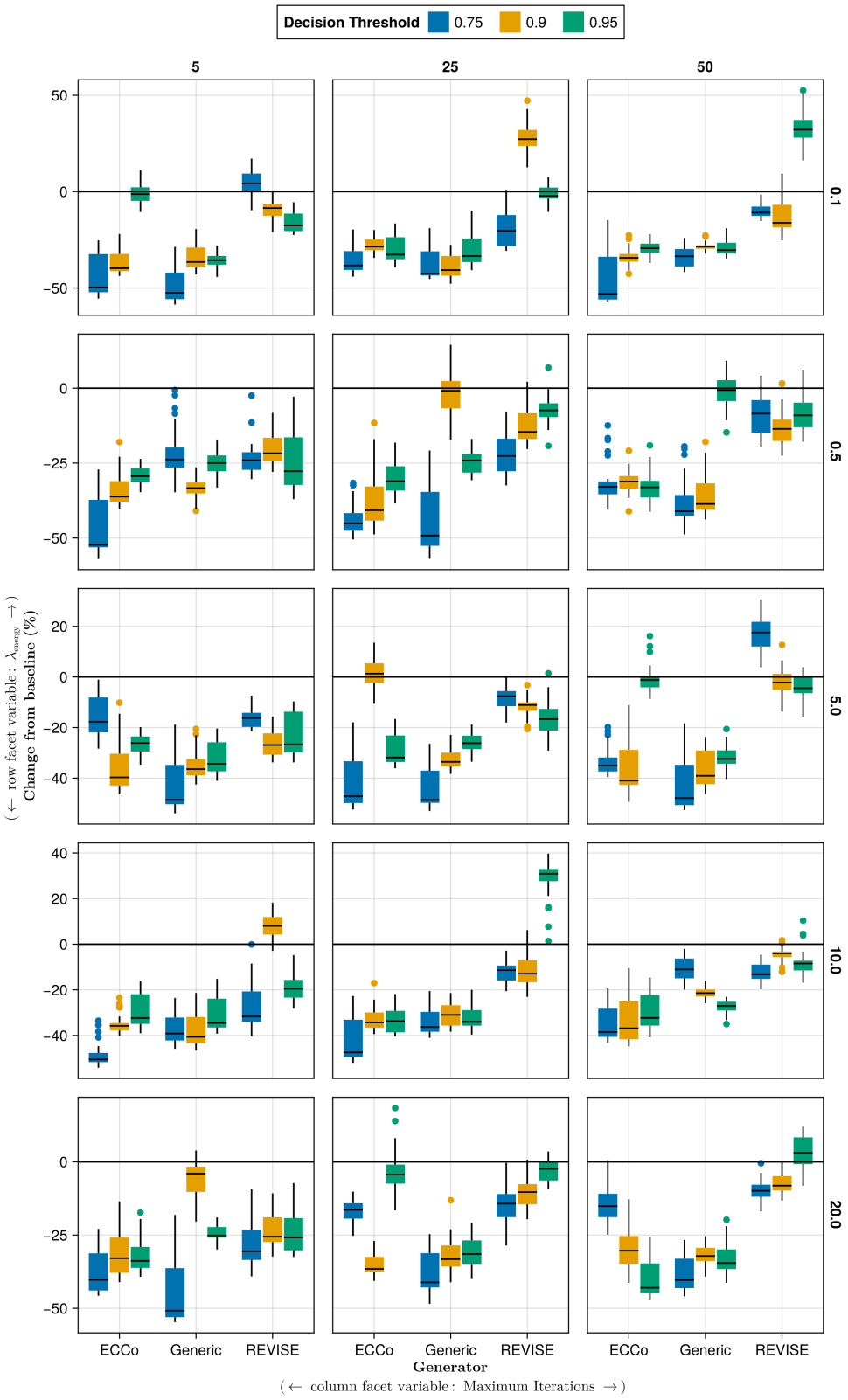

Figure 12: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

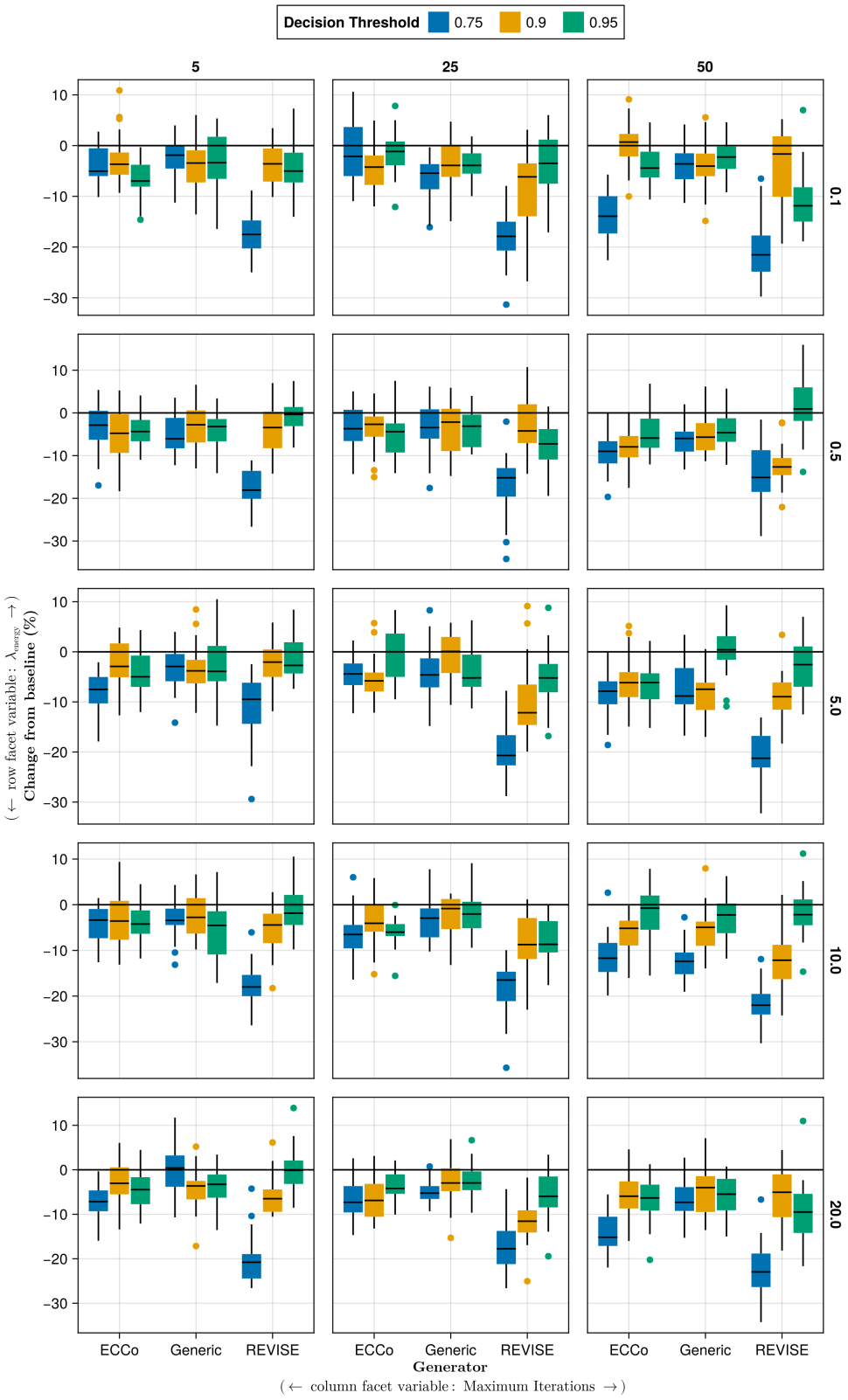

Figure 13: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

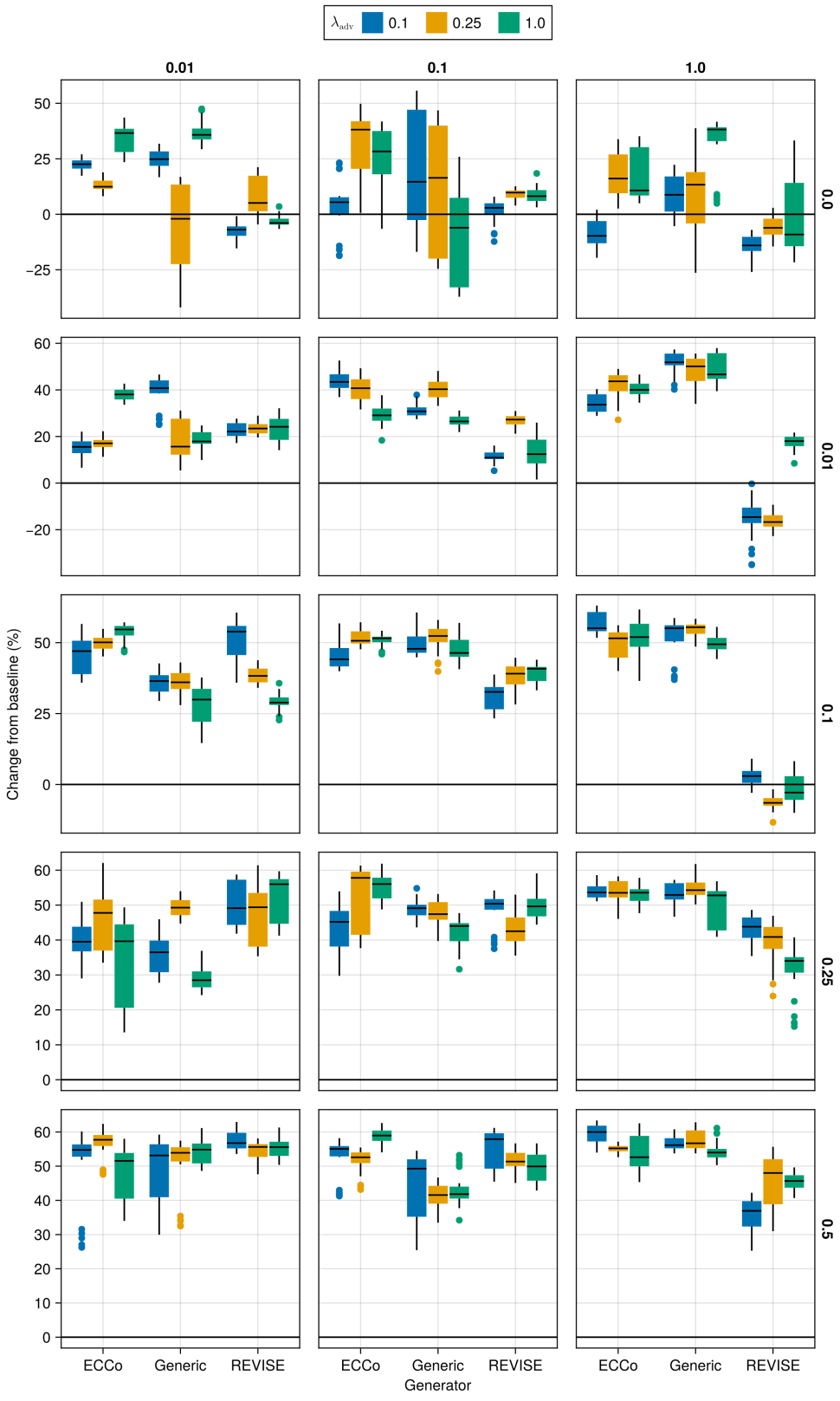

Figure 14: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

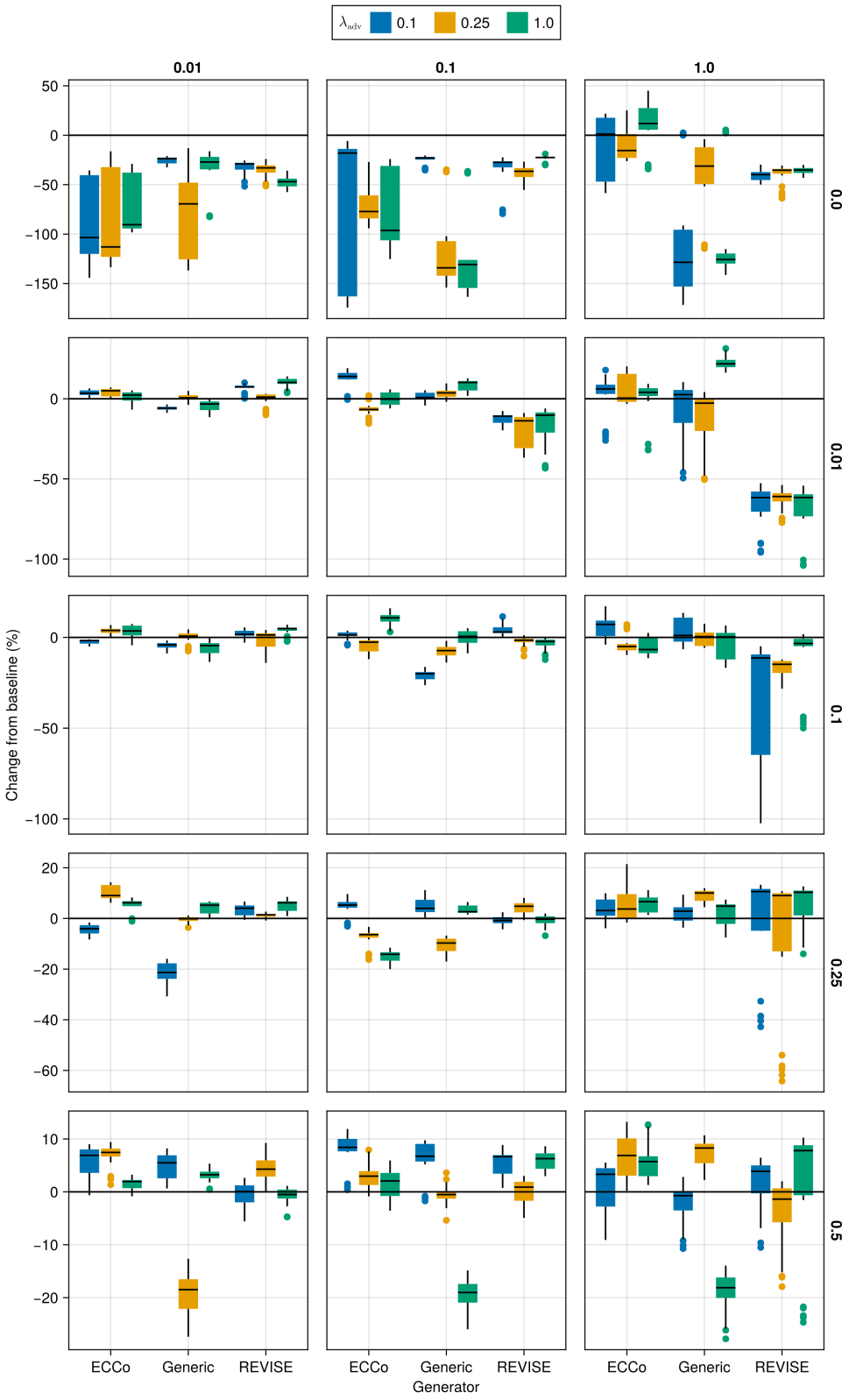

Figure 15: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

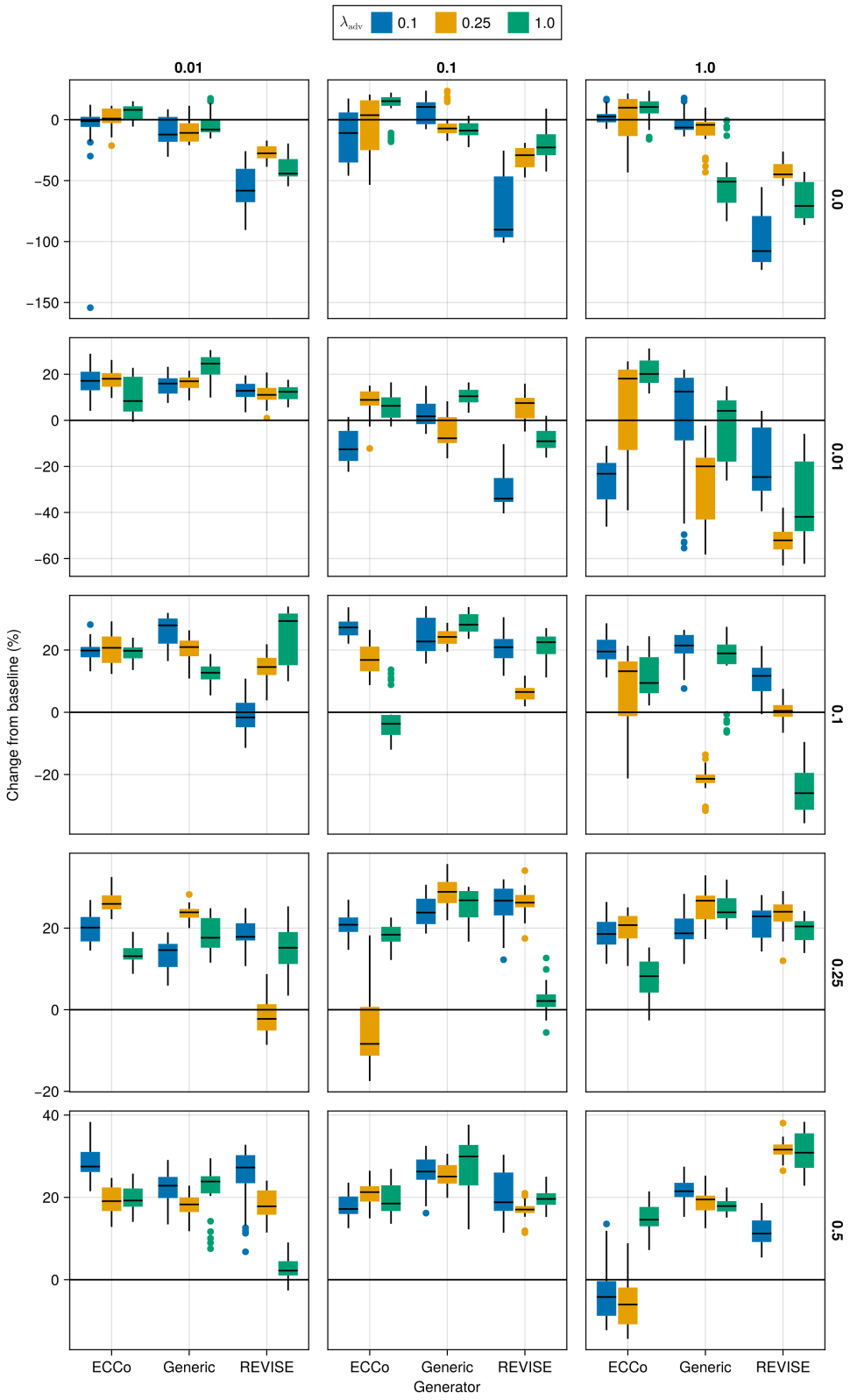

Figure 16: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

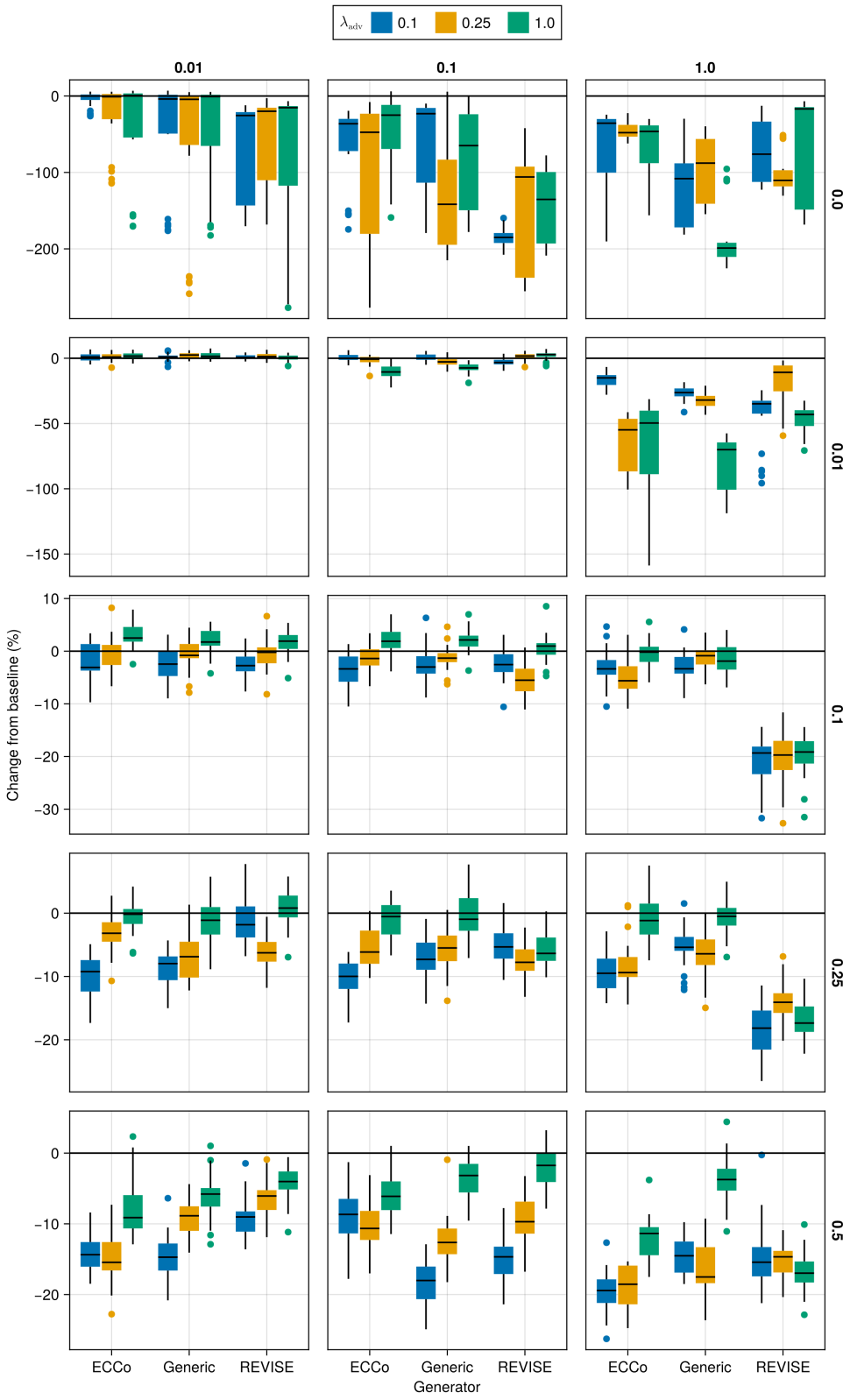

Figure 17: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

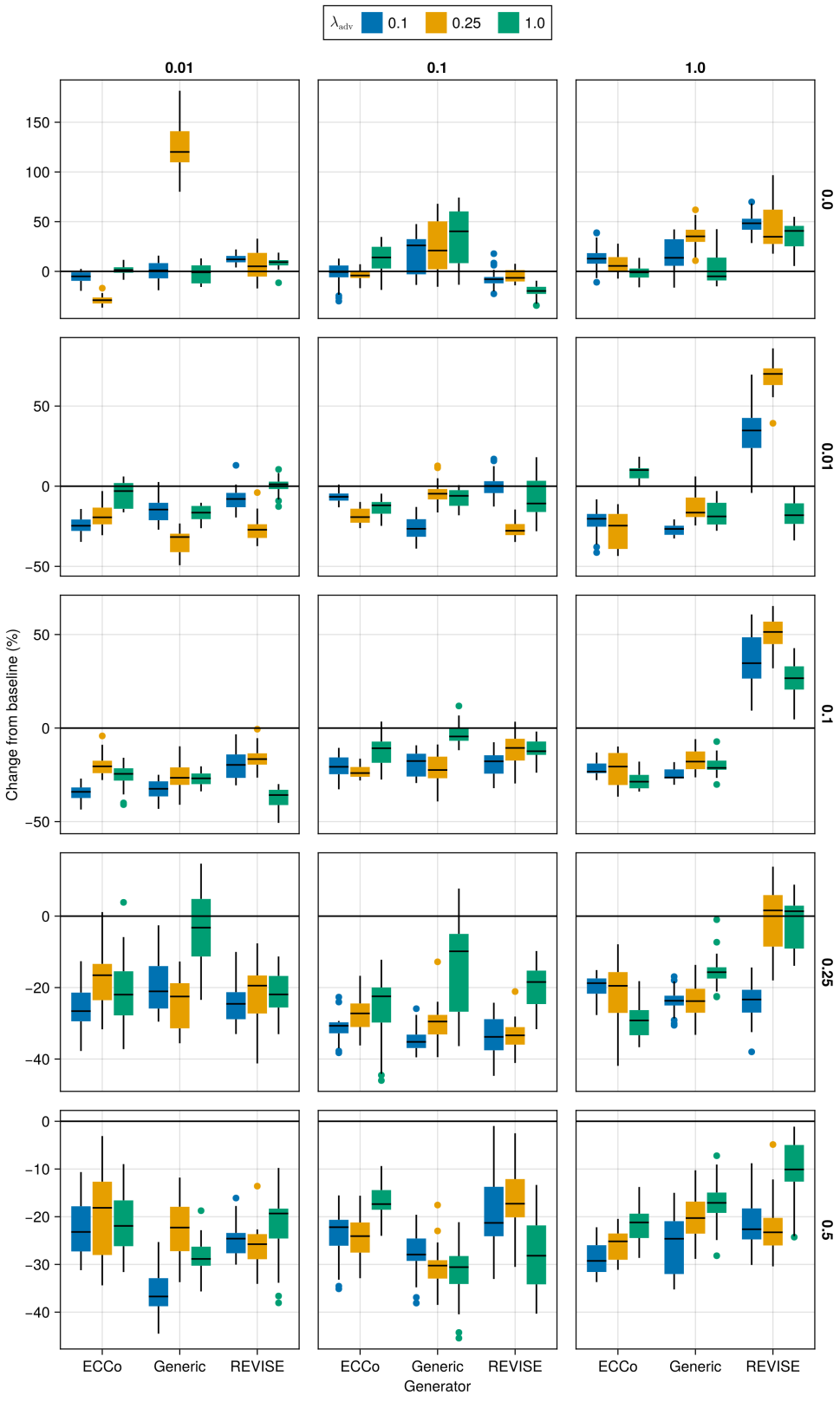

Figure 18: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

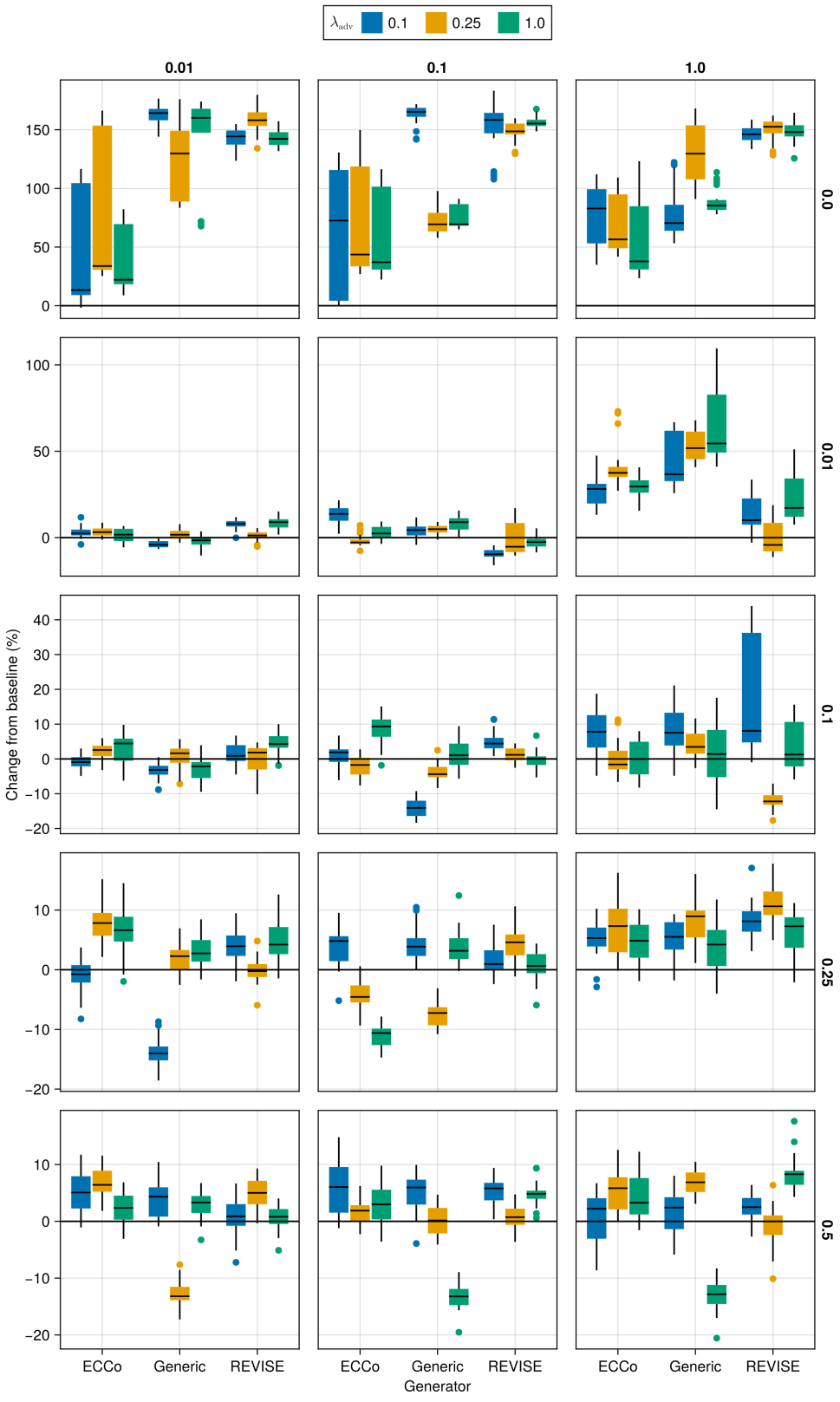

Figure 19: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

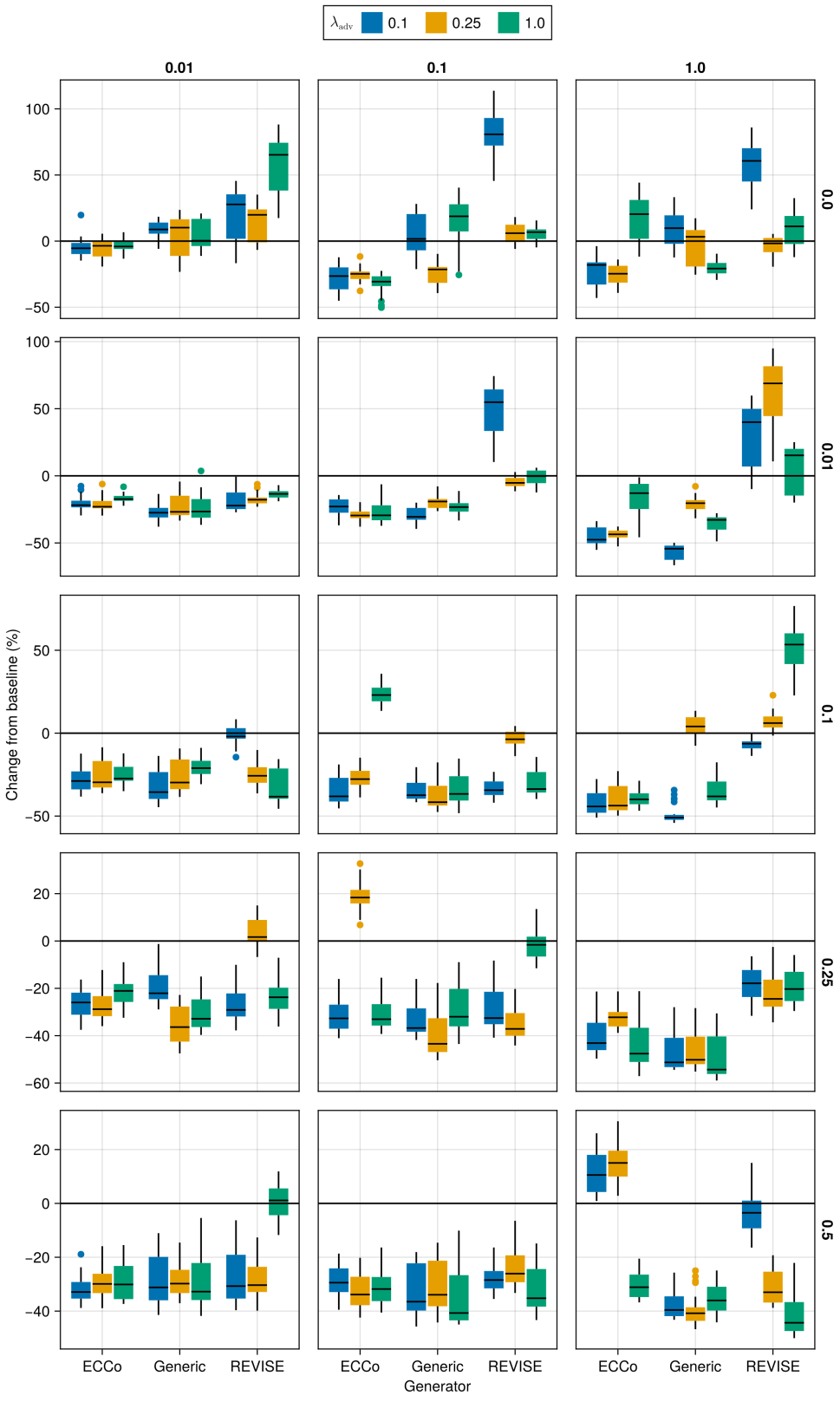

Figure 20: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

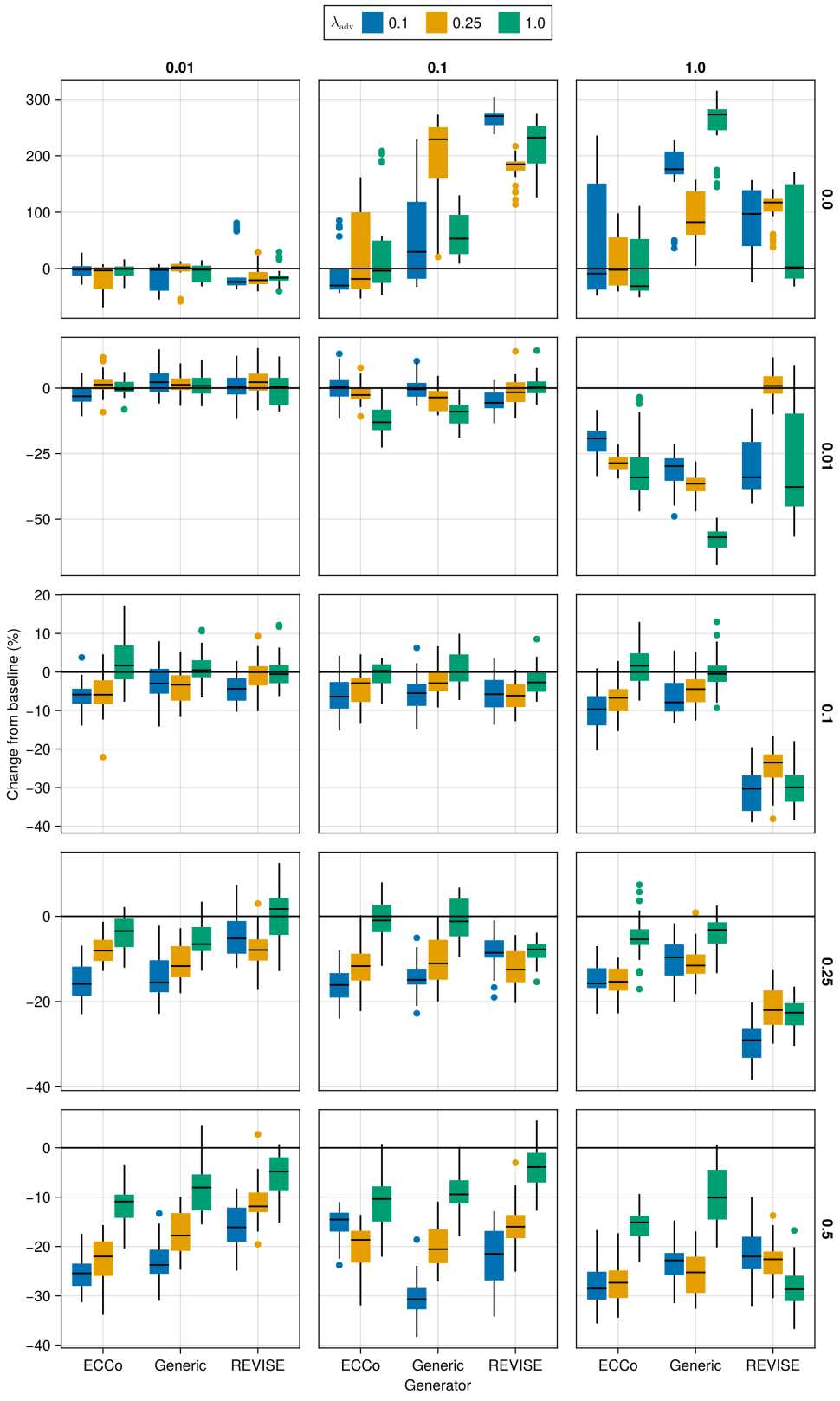

Figure 21: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

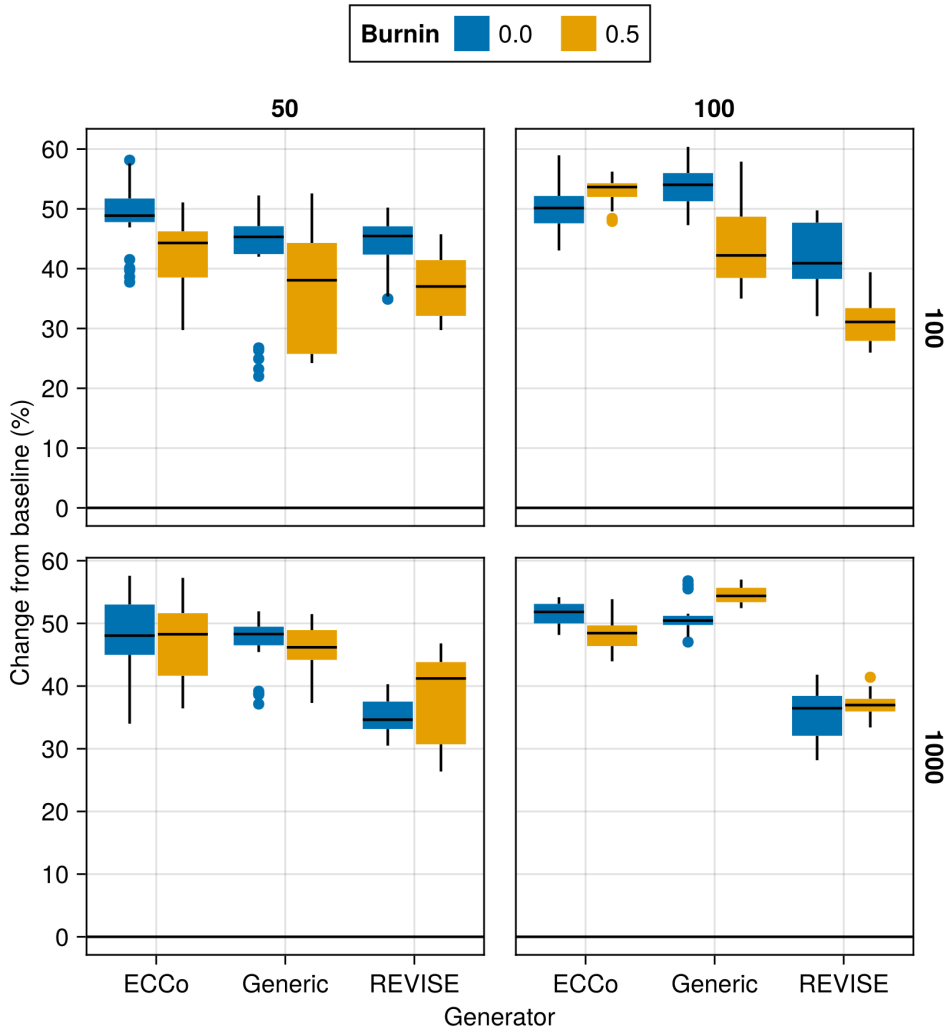

Figure 22: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

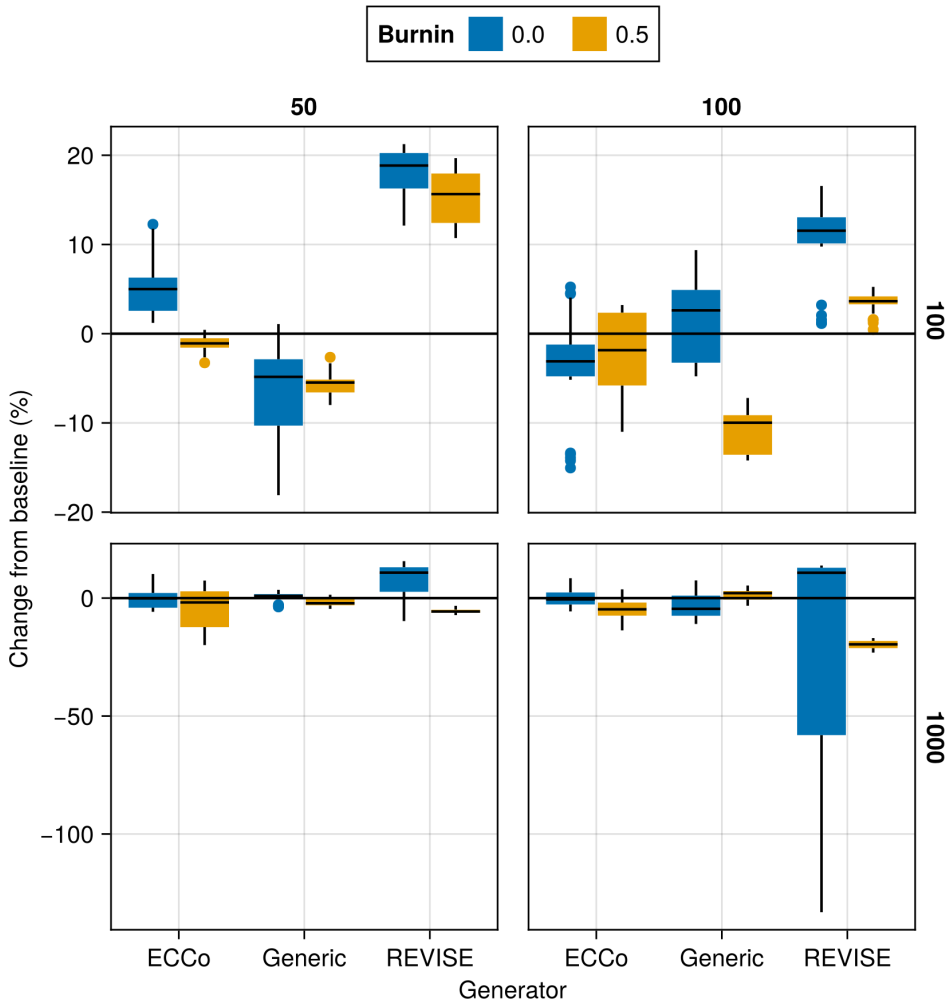

Figure 23: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

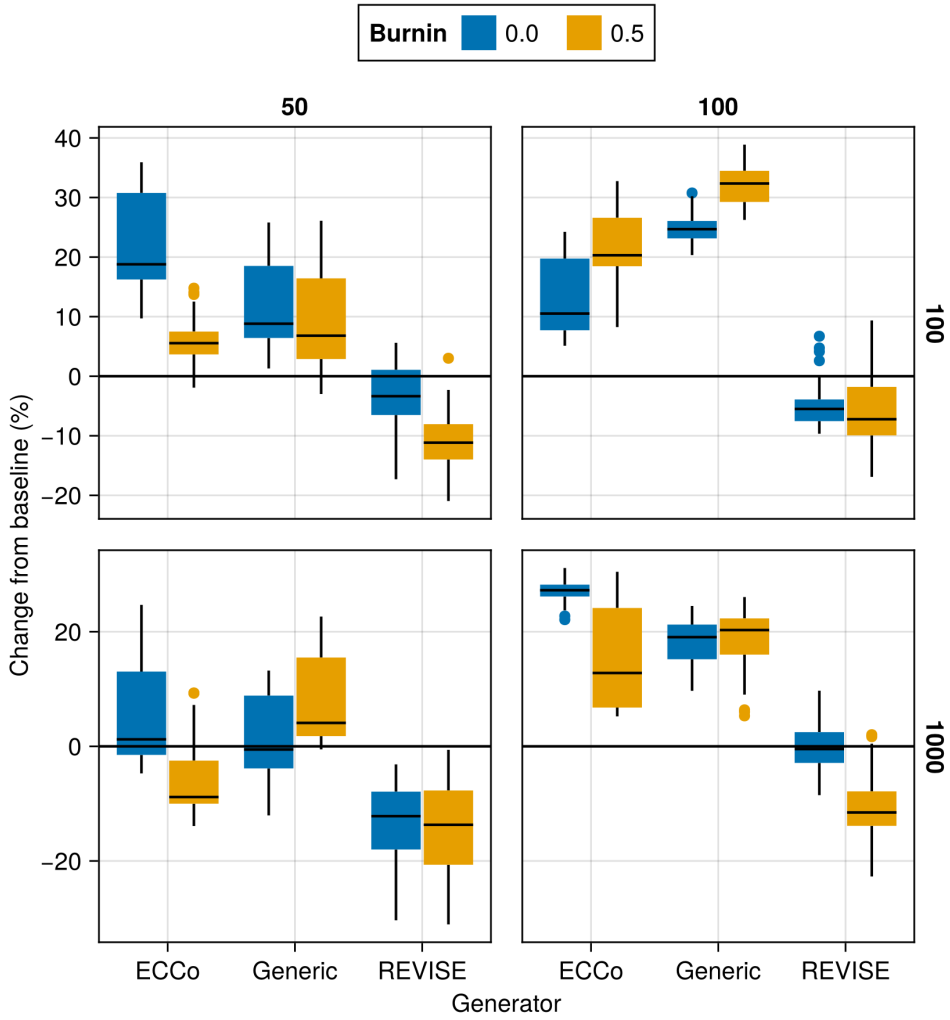

Figure 24: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

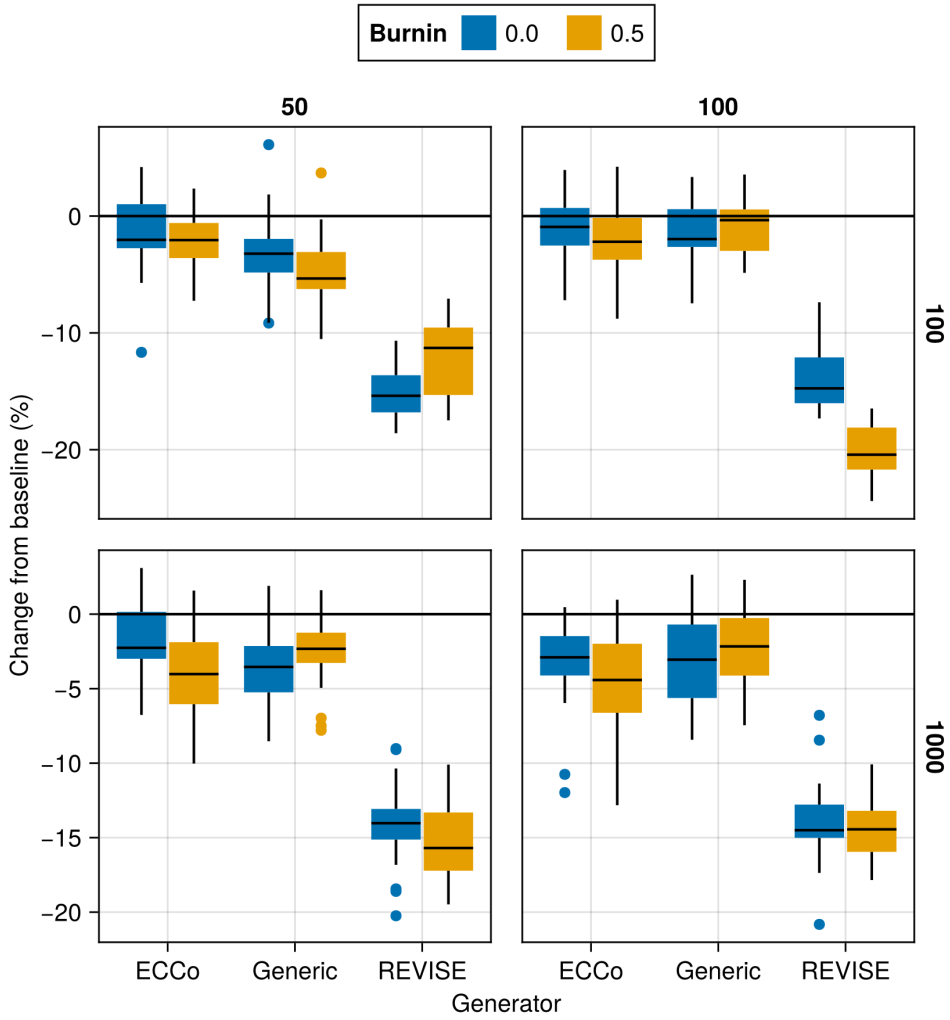

Figure 25: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

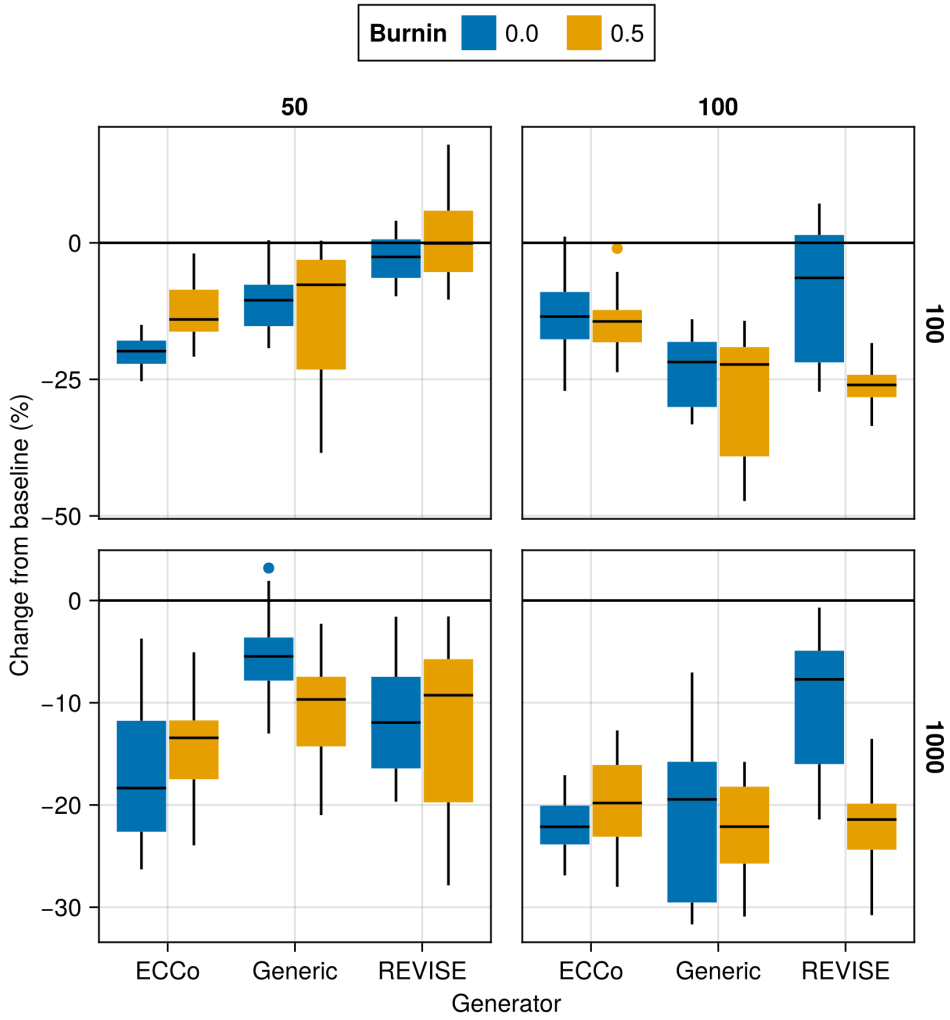

Figure 26: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

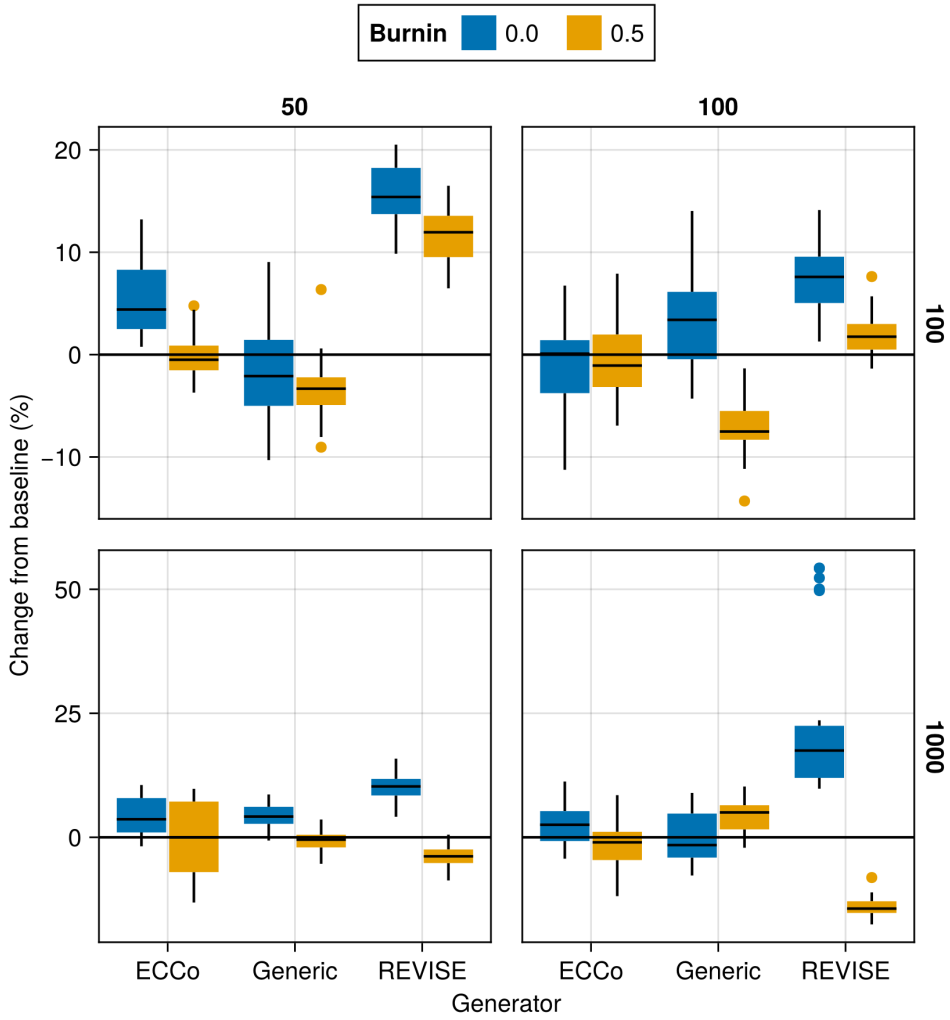

Figure 27: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

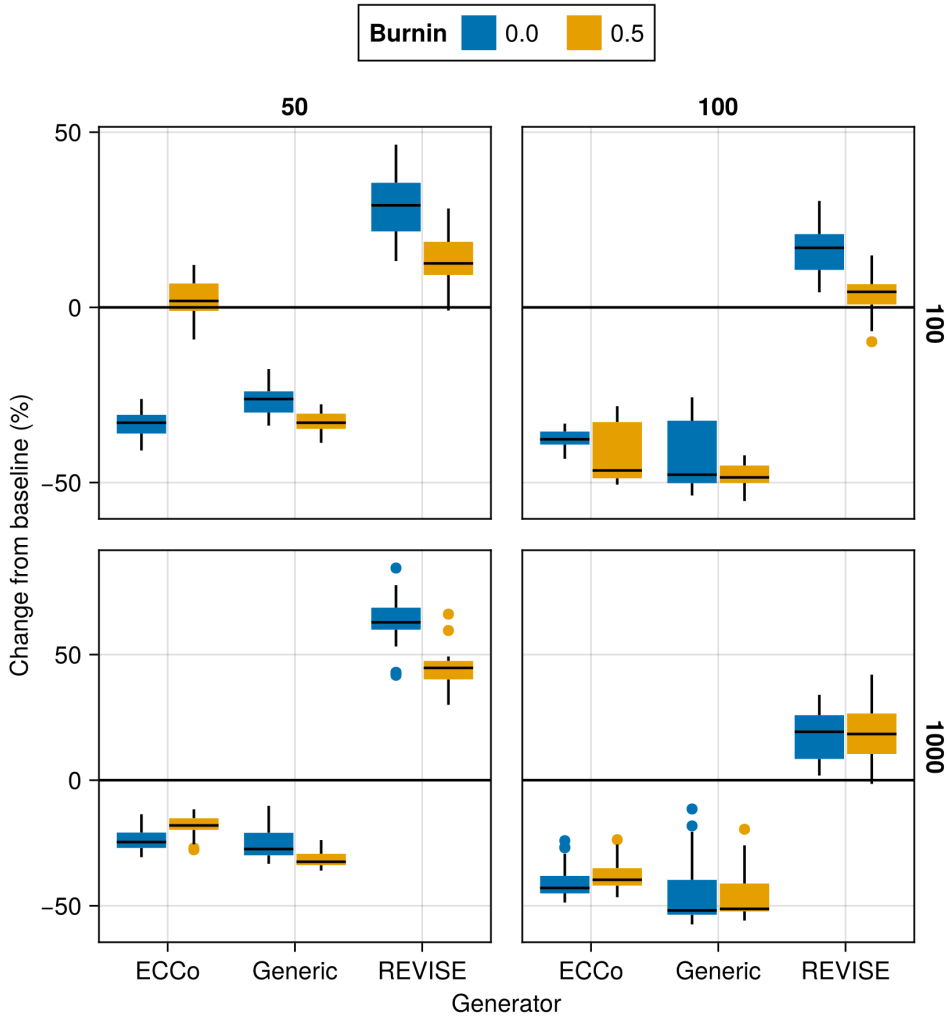

Figure 28: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

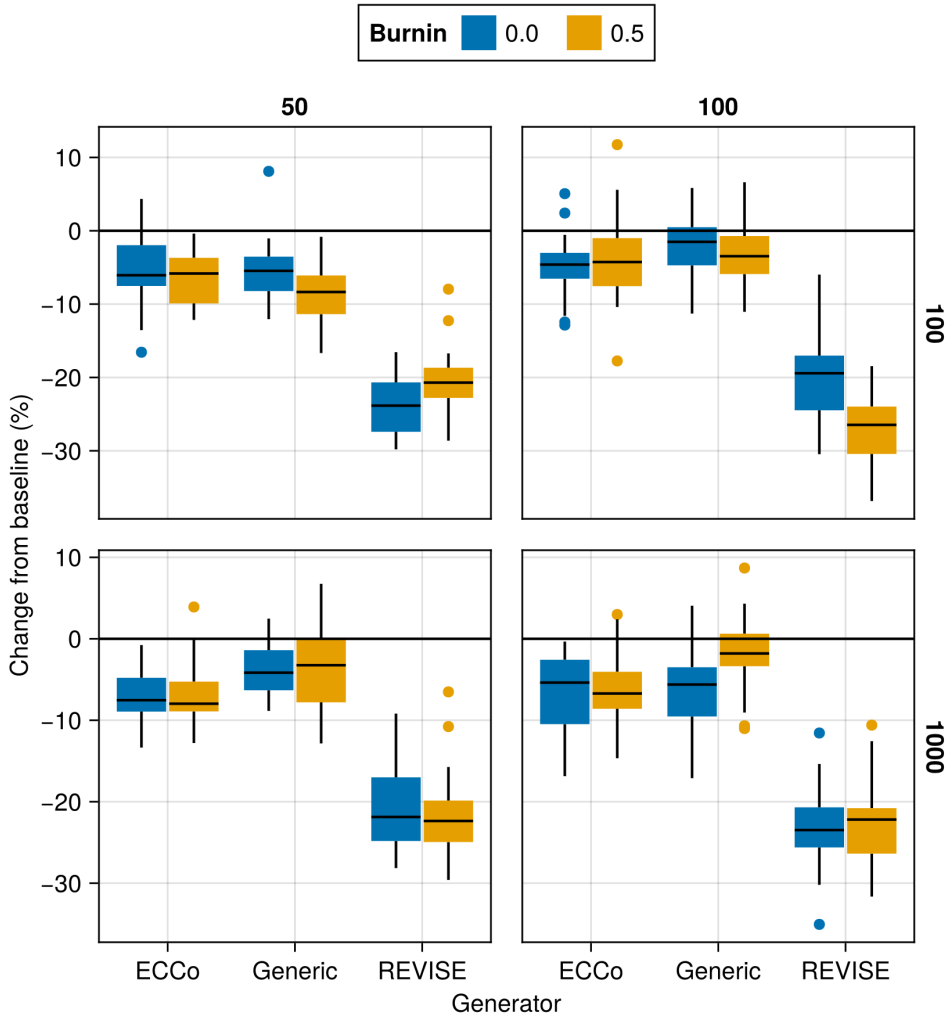

Figure 29: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

Appendix E Tuning Key Parameters

Based on the findings from our initial large grid searches (Section D), we tune selected hyperparameters for all datasets: namely, the decision threshold τ and the strength of the energy regularization λ_{reg}. The final hyperparameter choices for each dataset are presented in Table 2 in Section C. Detailed results for each data set are shown in Figure 30 to Figure 47. From Table 2, we notice that the same decision threshold of $\tau = 0.5$ is optimal for all but on dataset. We attribute this to the fact that a low decision threshold results in a higher share of mature counterfactuals and hence more opportunities for the model to learn from examples (Figure 39 to Figure 47). This has played a role in particular for our real-world tabular datasets and MNIST, which suffered from low levels of maturity for higher decision thresholds. In cases where maturity is not an issue, as for *Moons*, higher decision thresholds lead to better outcomes, which may have to do with the fact that the resulting counterfactuals are more faithful to the model. Concerning the regularization strength, we find somewhat high variation across datasets. Most notably, we find that relatively low levels of regularization are optimal for MNIST. We hypothesize that this finding may be attributed to the uniform scaling of all input features (digits).

Finally, to increase the proportion of mature counterfactuals for some datasets, we have also investigated the effect on the learning rate η for the counterfactual search and even smaller regularization strengths for a fixed decision threshold of 0.5 (Figure 48 to Figure 56). For the given low decision threshold, we find that the learning rate has no discernable impact on the proportion of mature counterfactuals (Figure 57 to Figure 65). We do notice, however, that the results for MNIST are much improved when using a low value λ_{reg}, the strength for the engery regularization: plausibility is increased by up to ~10% (Figure 54) and the proportion of mature counterfactuals reaches 100%.

One consideration worth exploring is to combine high decision thresholds with high learning rates, which we have not investigated here.

E.1 Key Parameters

The hyperparameter grid for tuning key parameters is shown in Note 9. The corresponding evaluation grid used for these experiments is shown in Note 10.

Note 9: Training Phase

- Generator Parameters:
  - Decision Threshold: `0.5, 0.75, 0.9`
- Model: `mlp`
- Training Parameters:
  - λ_{reg}: `0.1, 0.25, 0.5`
  - Objective: `full, vanilla`

Note 10: Evaluation Phase

- Generator Parameters:
  - λ_{egy}: `0.1, 0.5, 1.0, 5.0, 10.0`

E.1.1 Plausibility

The results with respect to the plausibility measure are shown in Figure 30 to Figure 38.

E.1.2 Proportion of Mature CE

The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure 39 to Figure 47.

E.2 Learning Rate

The hyperparameter grid for tuning the learning rate is shown in Note 11. The corresponding evaluation grid used for these experiments is shown in Note 12.

Note 11: Training Phase

- Generator Parameters:
  - Learning Rate: `0.1, 0.5, 1.0`

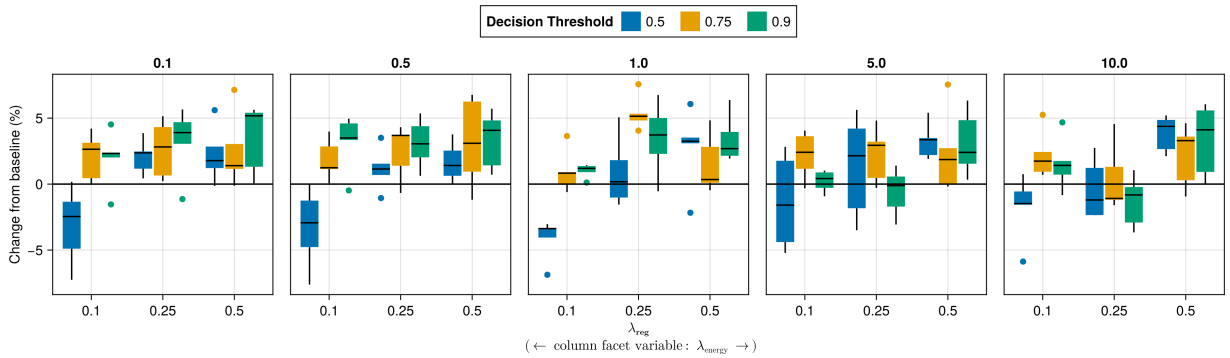

Figure 30: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Adult.

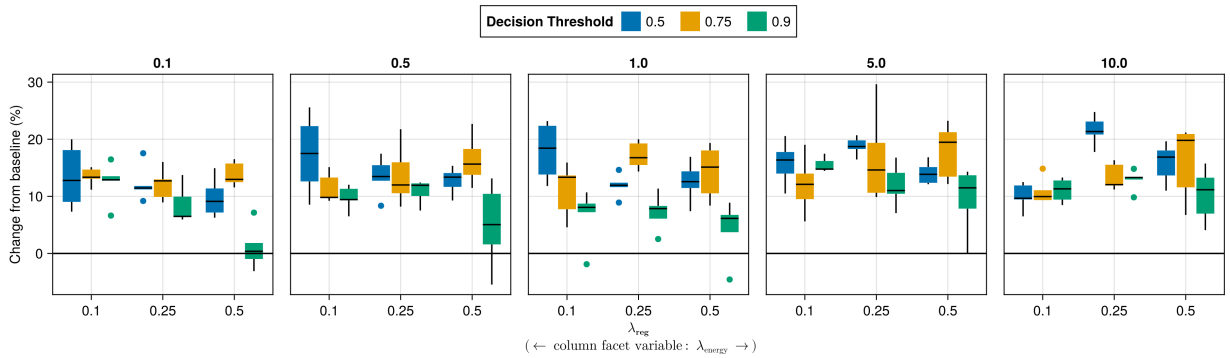

Figure 31: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: California Housing.

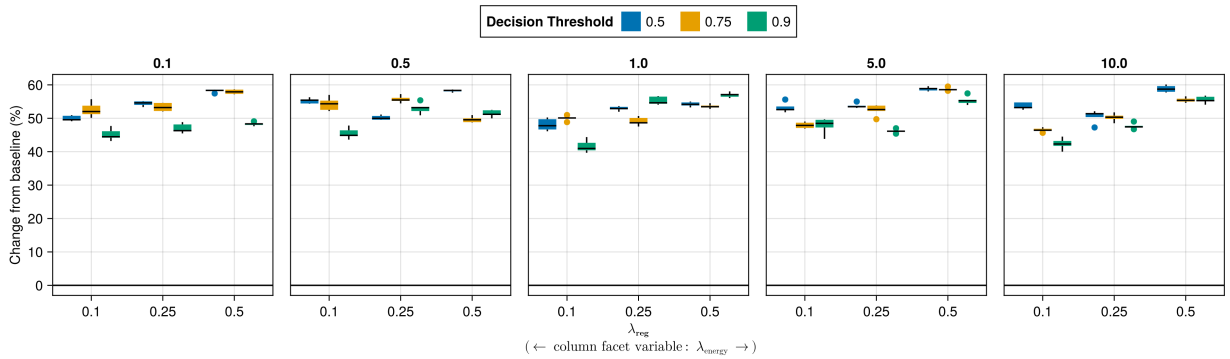

Figure 32: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

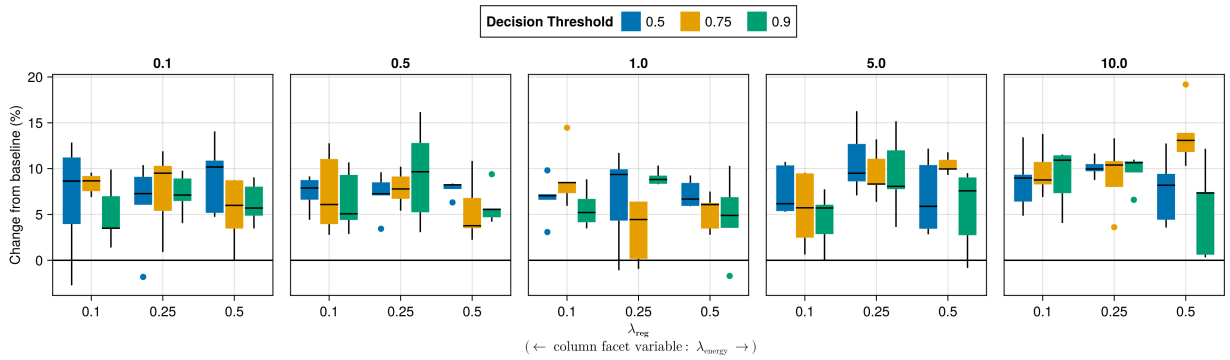

Figure 33: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Credit.

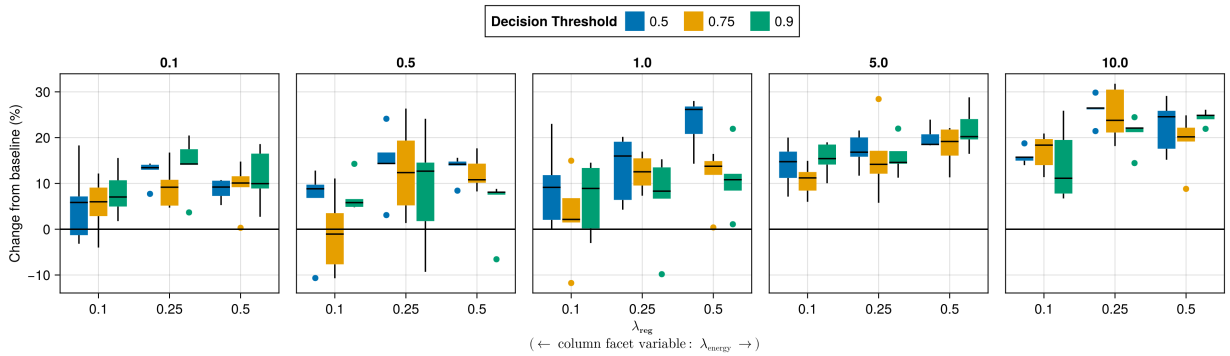

Figure 34: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: GMSC.

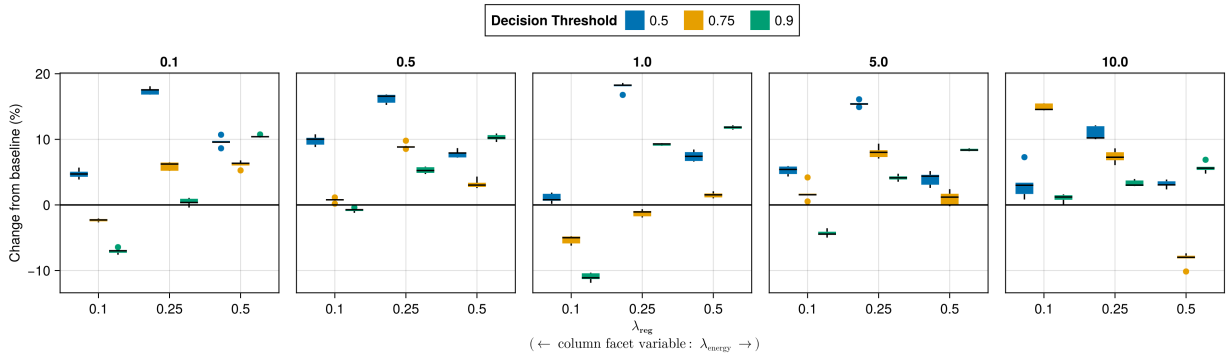

Figure 35: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

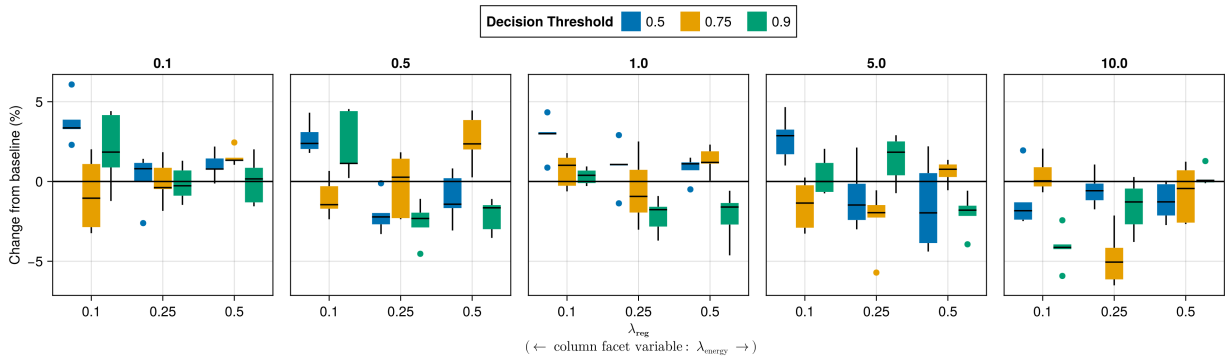

Figure 36: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: MNIST.

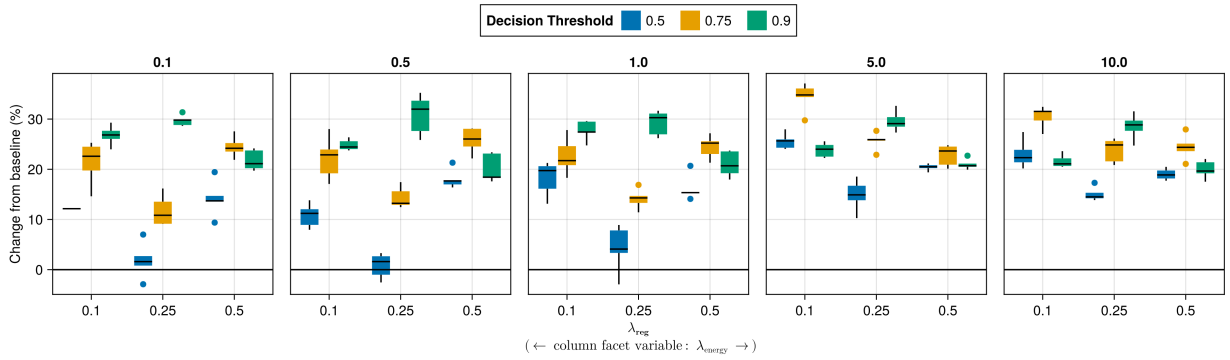

Figure 37: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

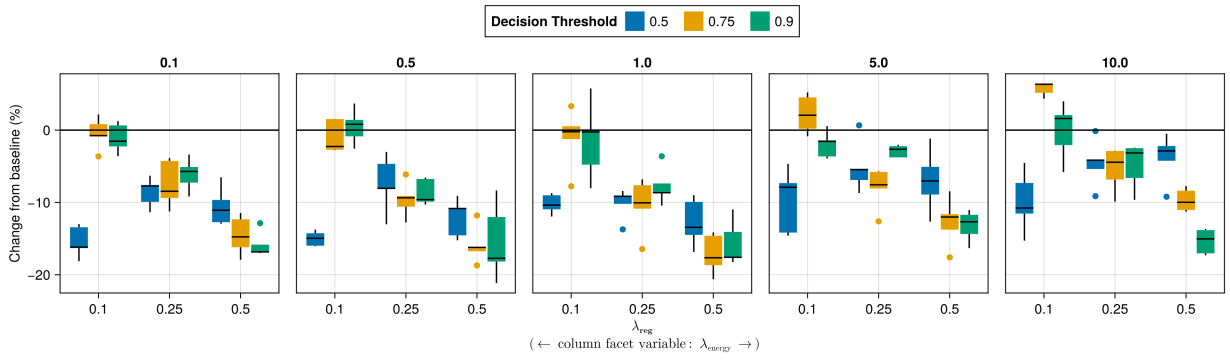

Figure 38: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

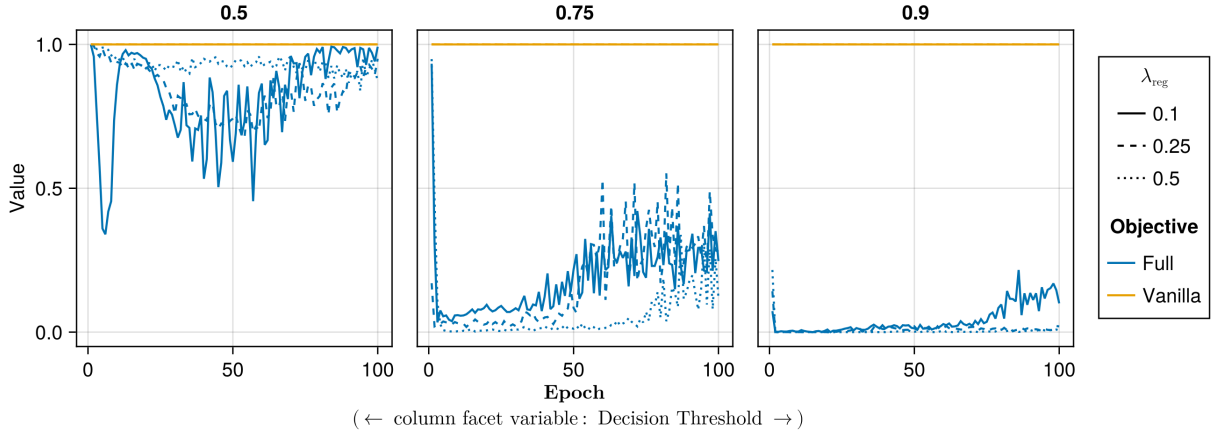

Figure 39: Proportion of mature counterfactuals in each epoch. Data: Adult.

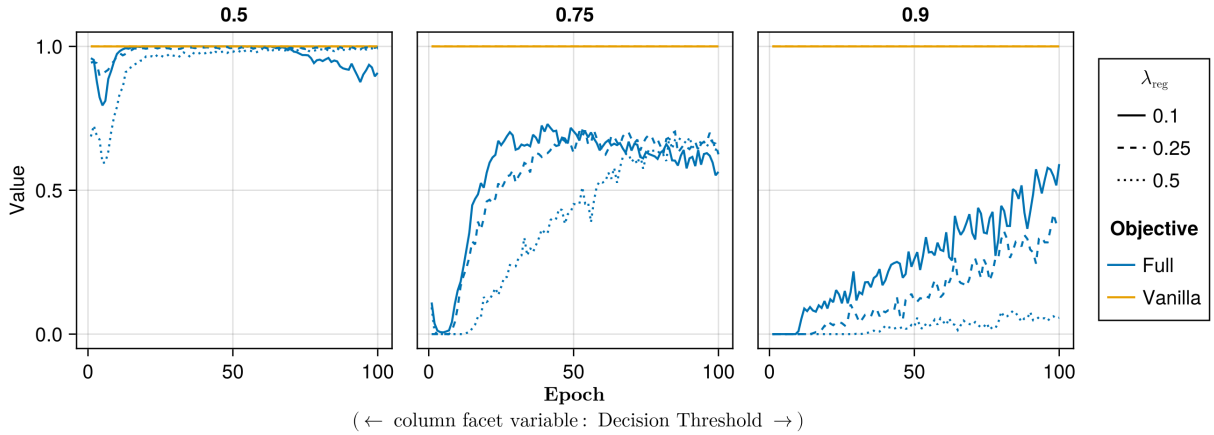

Figure 40: Proportion of mature counterfactuals in each epoch. Data: California Housing.

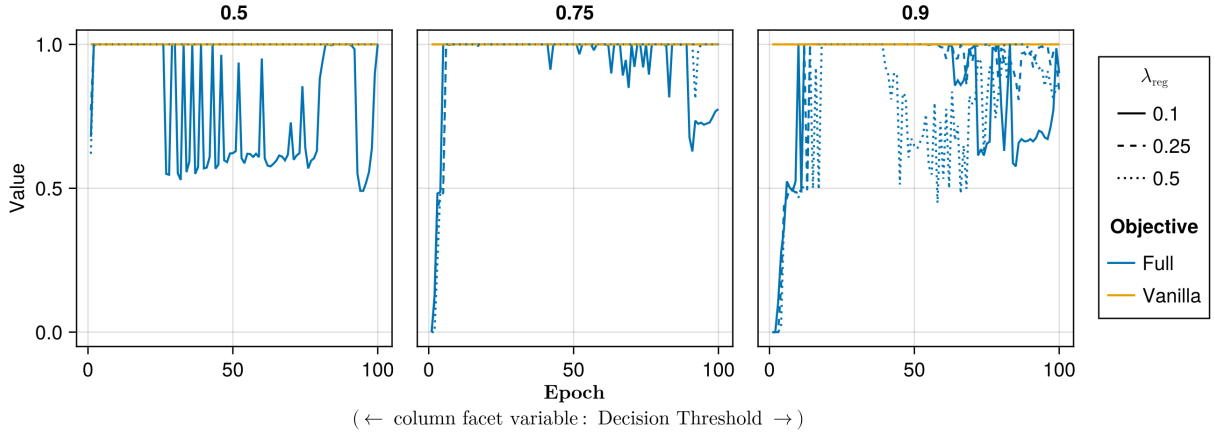

Figure 41: Proportion of mature counterfactuals in each epoch. Data: Circles.

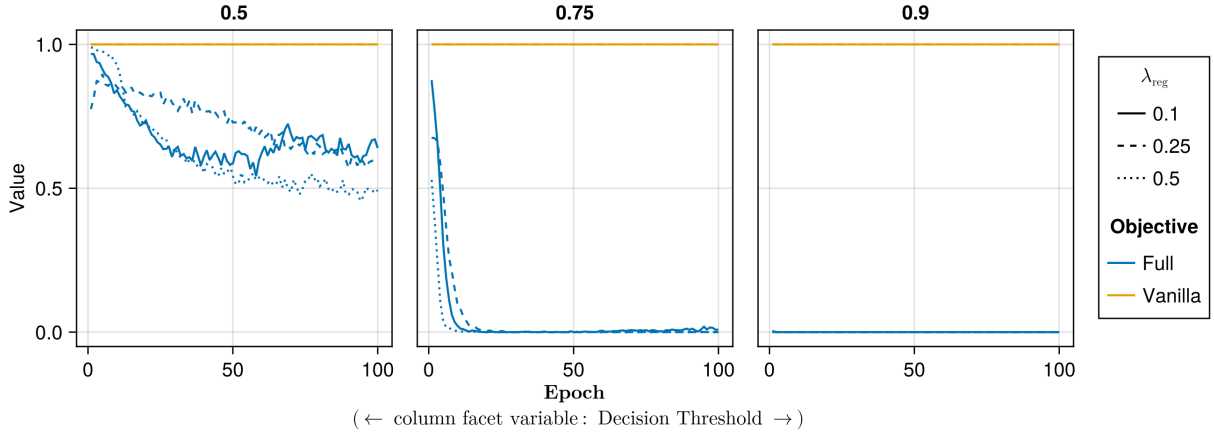

Figure 42: Proportion of mature counterfactuals in each epoch. Data: Credit.

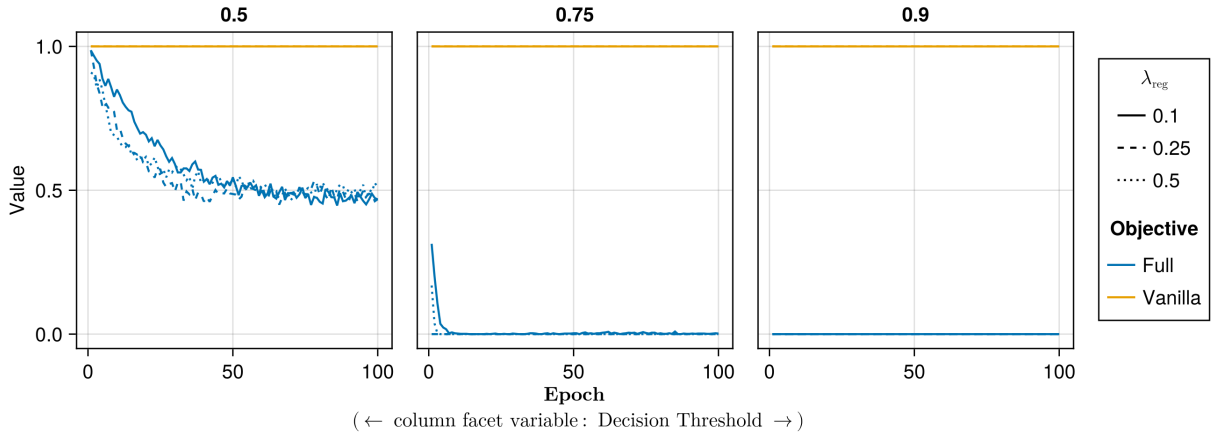

Figure 43: Proportion of mature counterfactuals in each epoch. Data: GMSC.

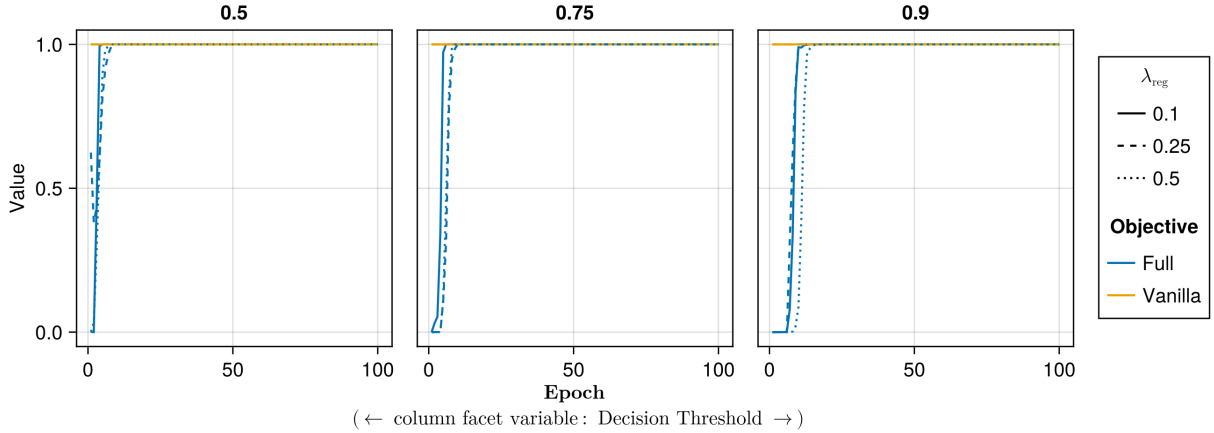

Figure 44: Proportion of mature counterfactuals in each epoch. Data: Linearly Separable.

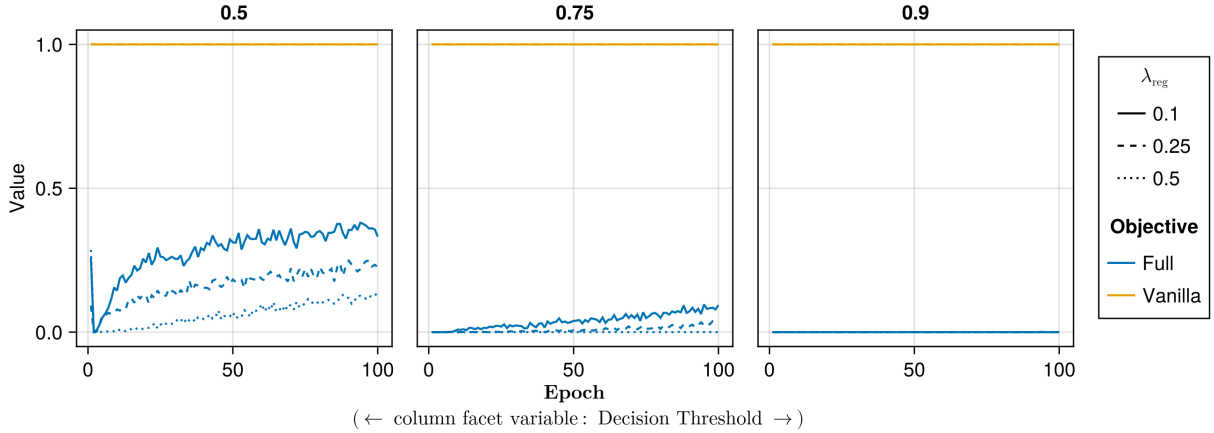

Figure 45: Proportion of mature counterfactuals in each epoch. Data: MNIST.

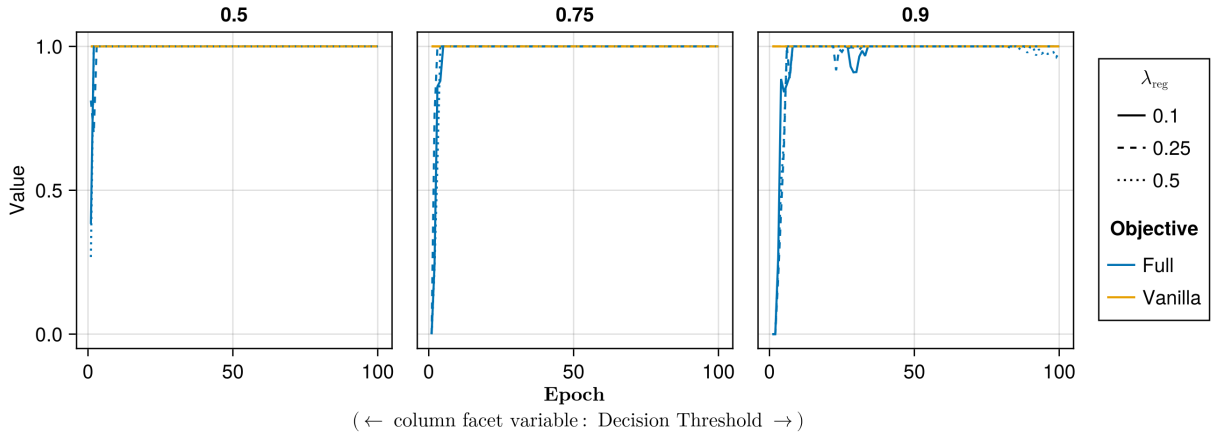

Figure 46: Proportion of mature counterfactuals in each epoch. Data: Moons.

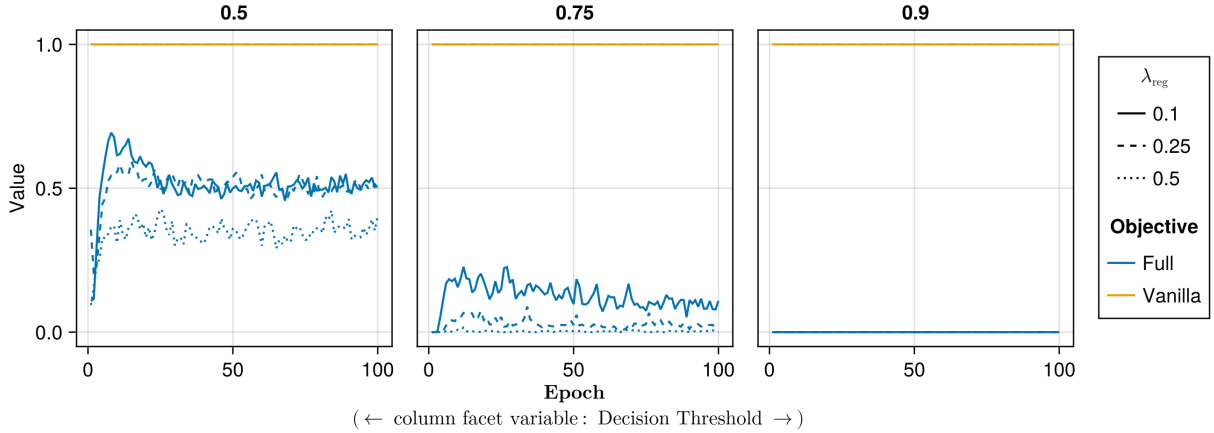

Figure 47: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

- Model: `mlp`
- Training Parameters:
  - λ_{reg}: `0.01, 0.1, 0.5`
  - Objective: `full, vanilla`

Note 12: Evaluation Phase

- Generator Parameters:
  - λ_{egy}: `0.1, 0.5, 1.0, 5.0, 10.0`

E.2.1 Plausibility

The results with respect to the plausibility measure are shown in Figure 48 to Figure 56.

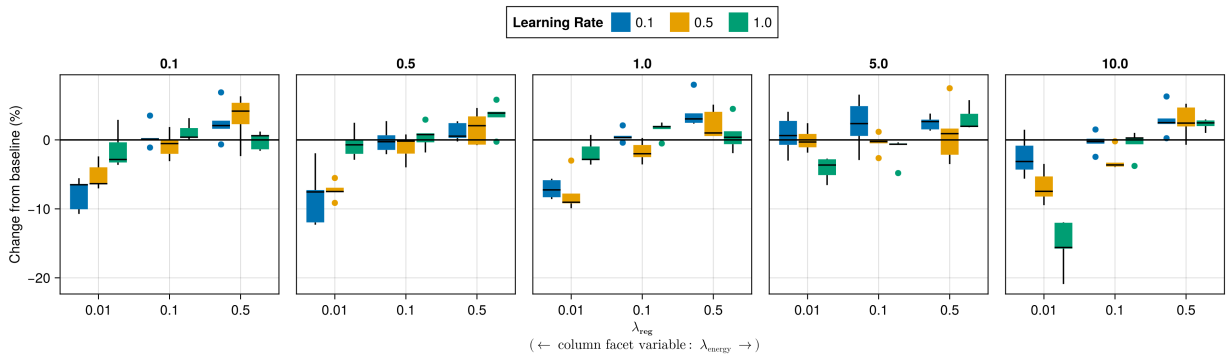

Figure 48: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Adult.

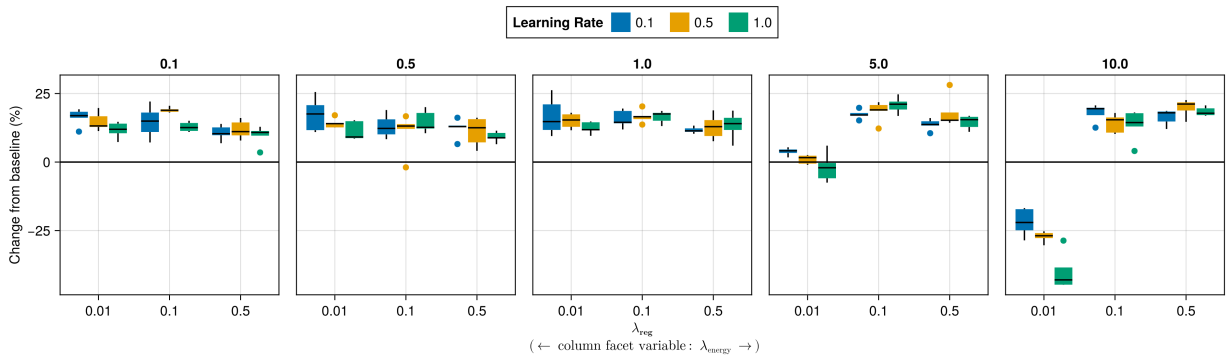

Figure 49: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: California Housing.

E.2.2 Proportion of Mature CE

The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure 57 to Figure 65.

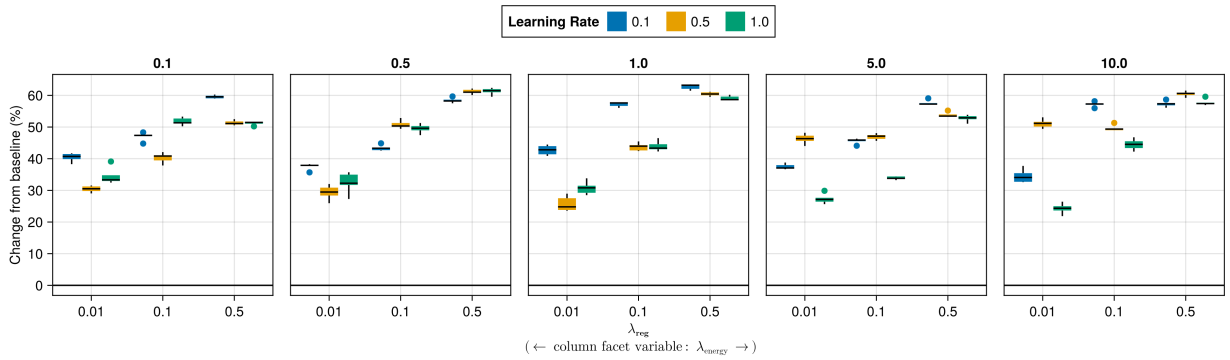

Figure 50: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

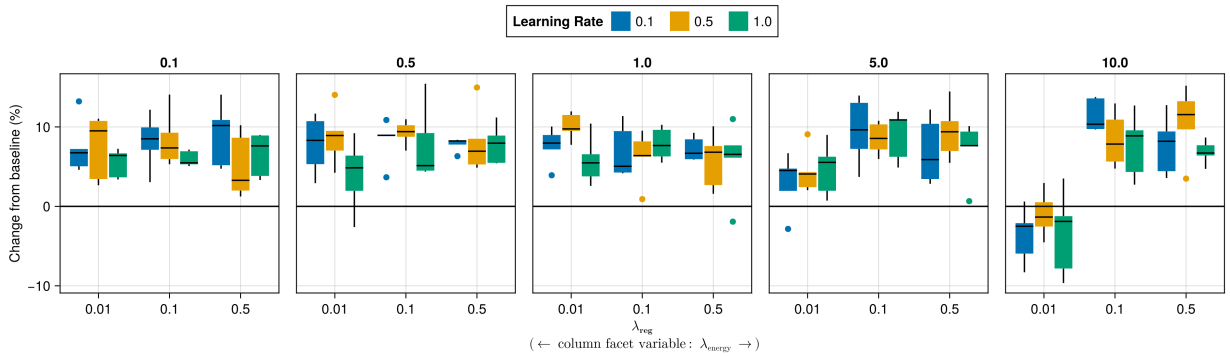

Figure 51: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Credit.

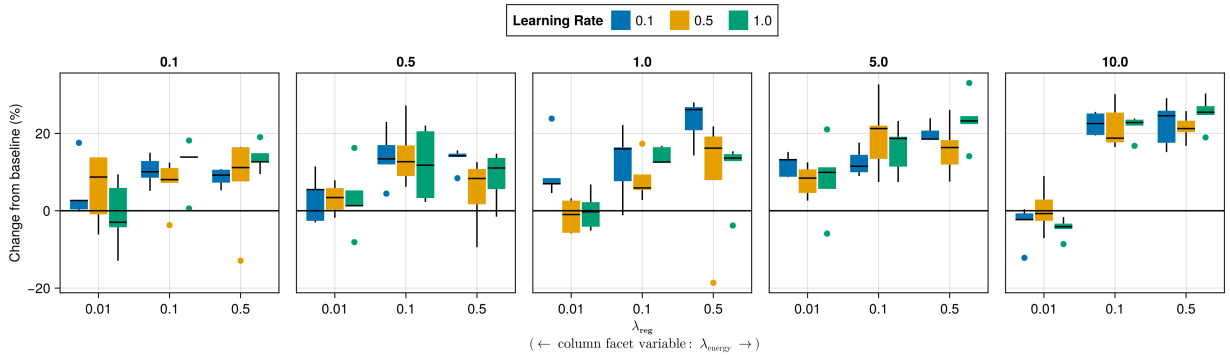

Figure 52: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: GMSC.

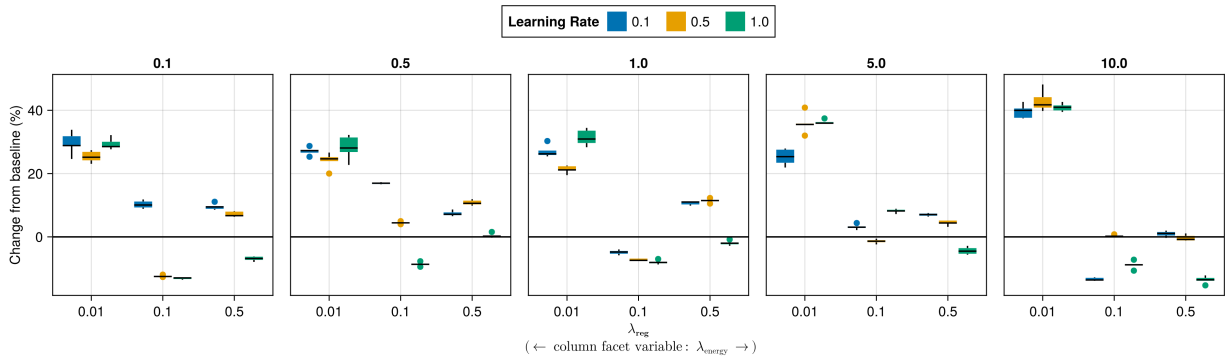

Figure 53: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

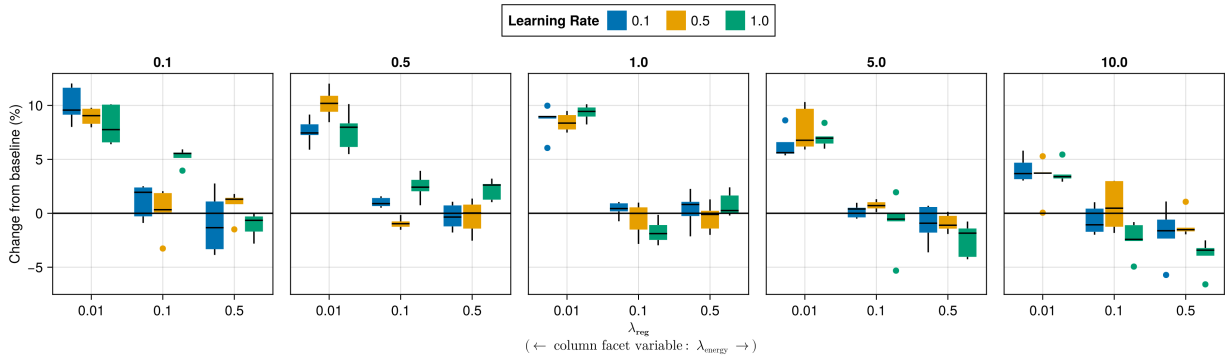

Figure 54: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: MNIST.

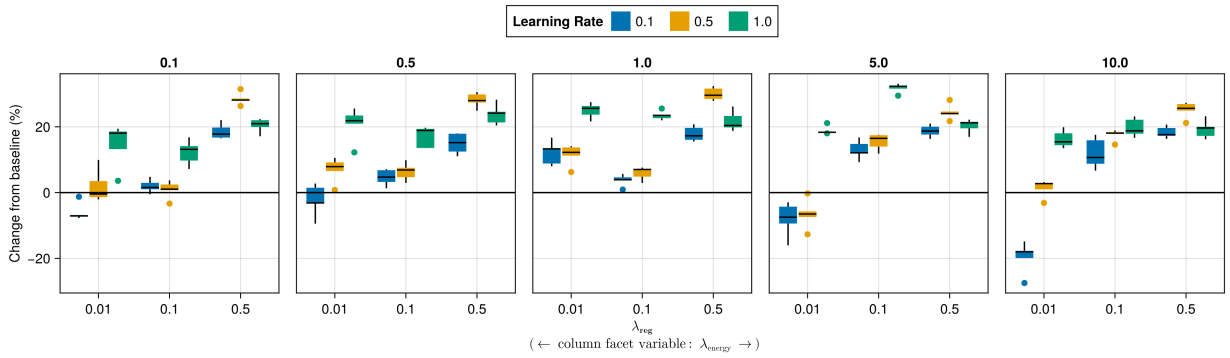

Figure 55: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

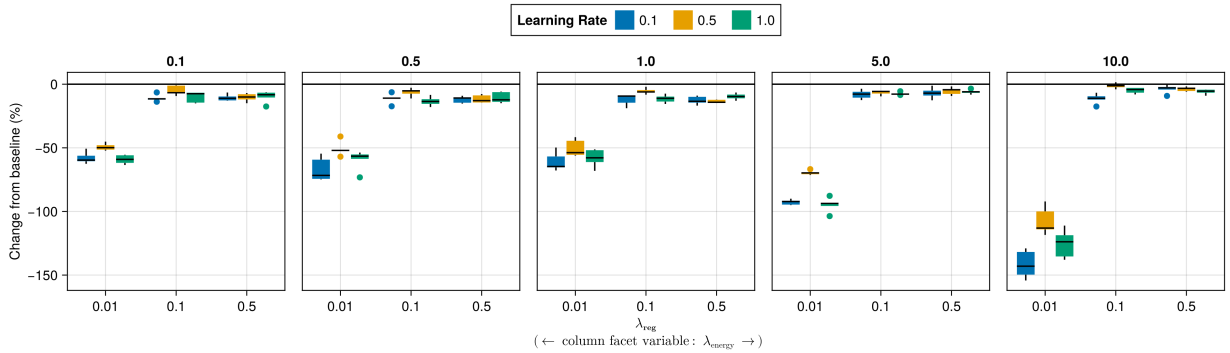

Figure 56: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 5). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

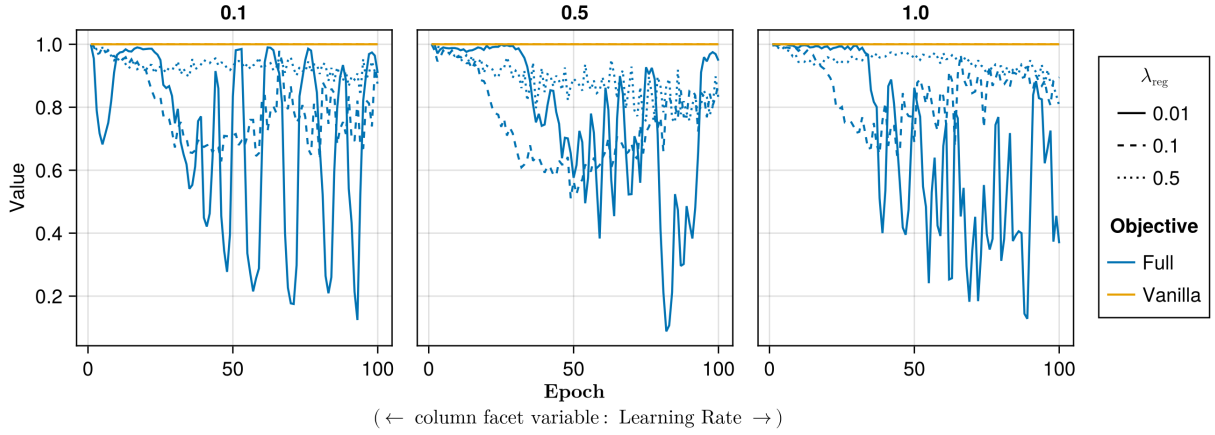

Figure 57: Proportion of mature counterfactuals in each epoch. Data: Adult.

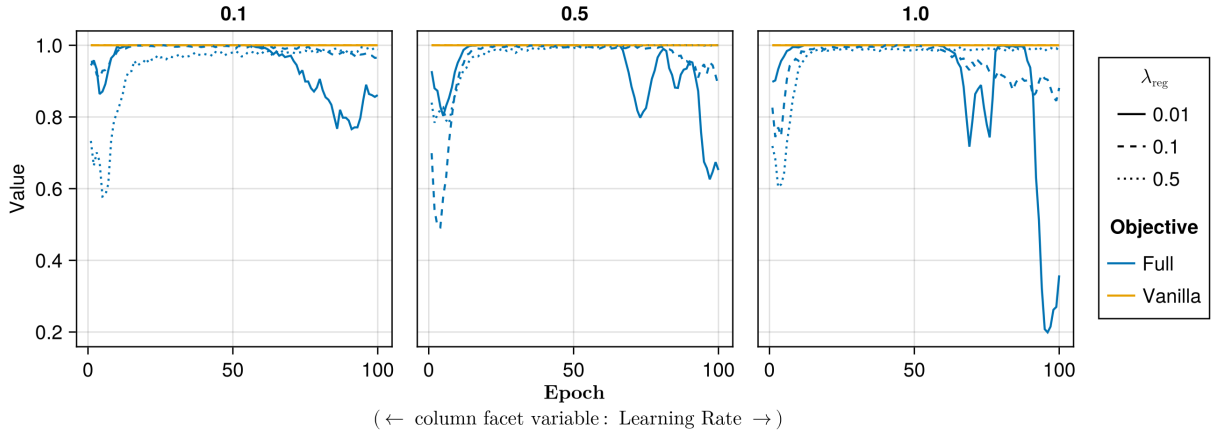

Figure 58: Proportion of mature counterfactuals in each epoch. Data: California Housing.

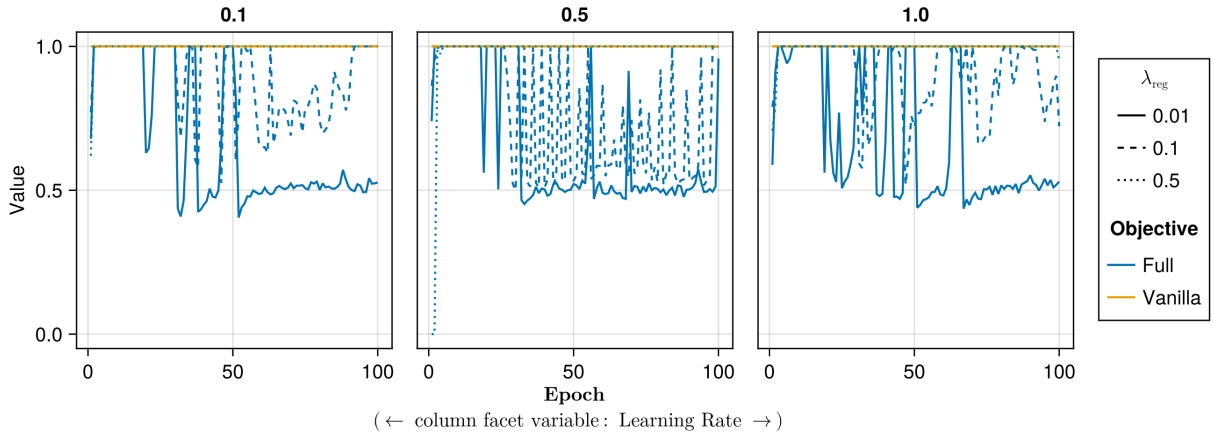

Figure 59: Proportion of mature counterfactuals in each epoch. Data: Circles.

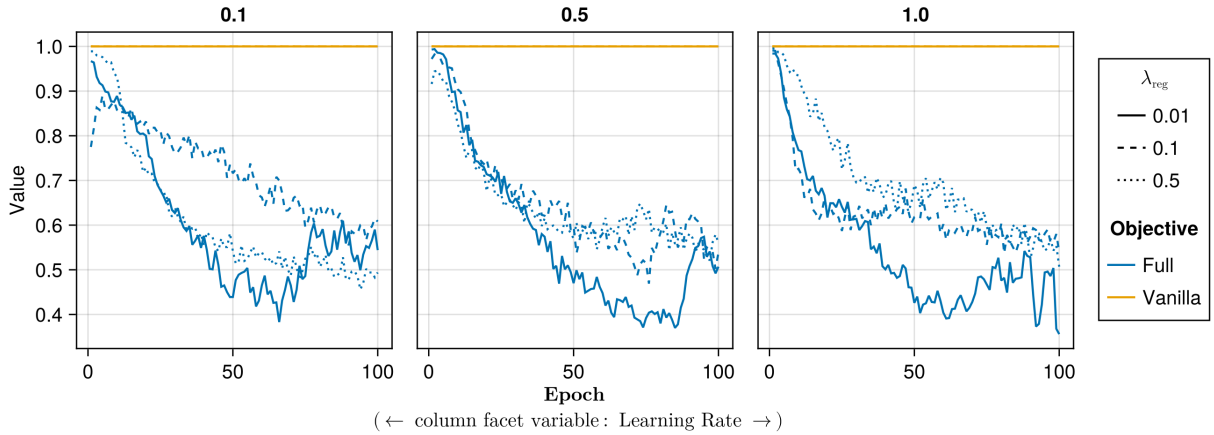

Figure 60: Proportion of mature counterfactuals in each epoch. Data: Credit.

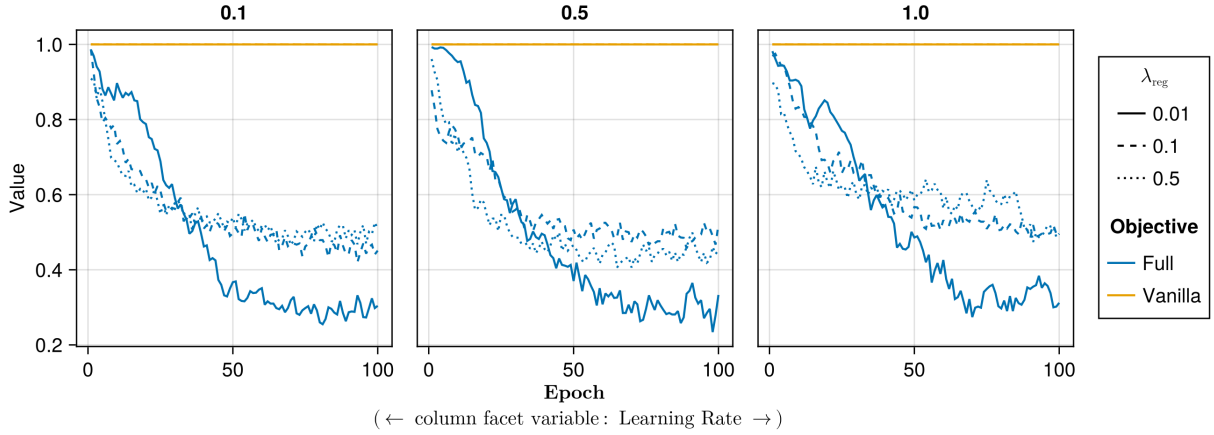

Figure 61: Proportion of mature counterfactuals in each epoch. Data: GMSC.

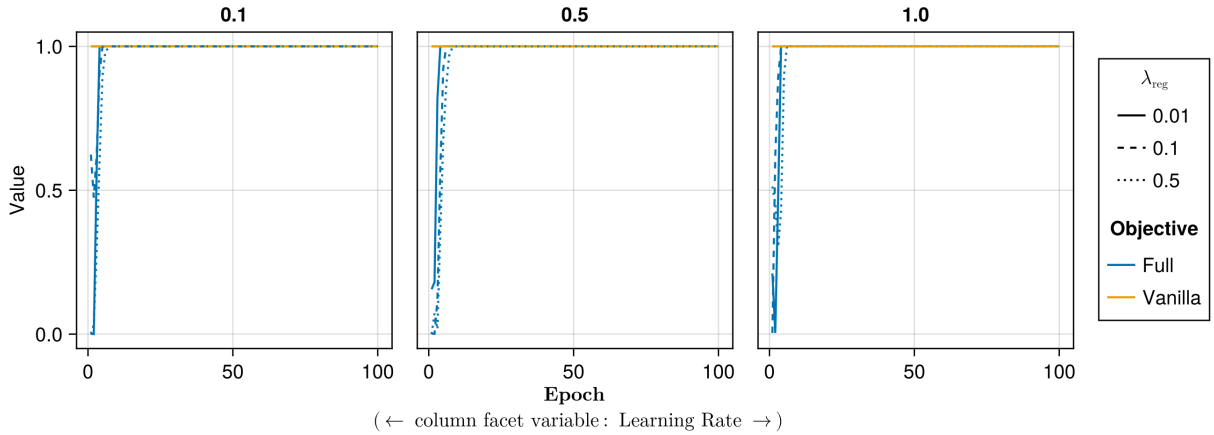

Figure 62: Proportion of mature counterfactuals in each epoch. Data: Linearly Separable.

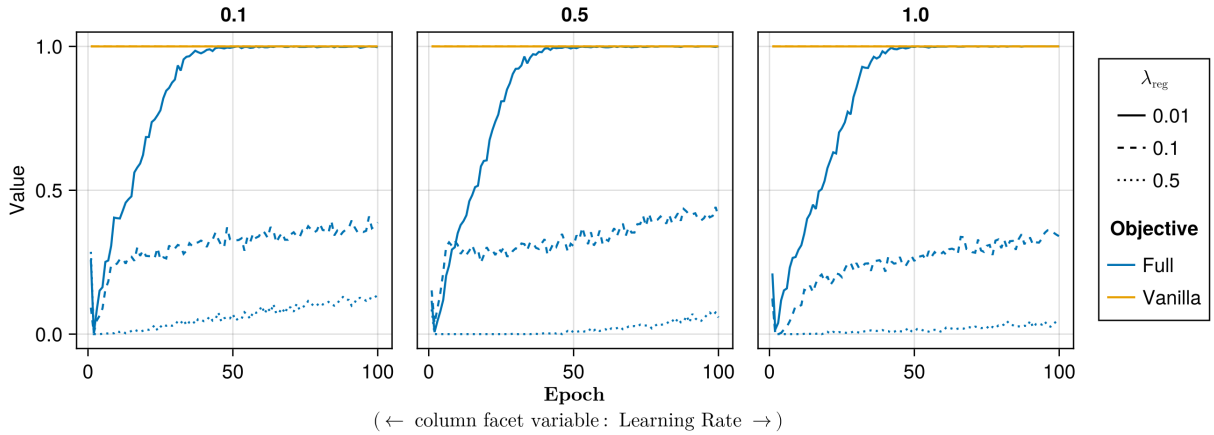

Figure 63: Proportion of mature counterfactuals in each epoch. Data: MNIST.

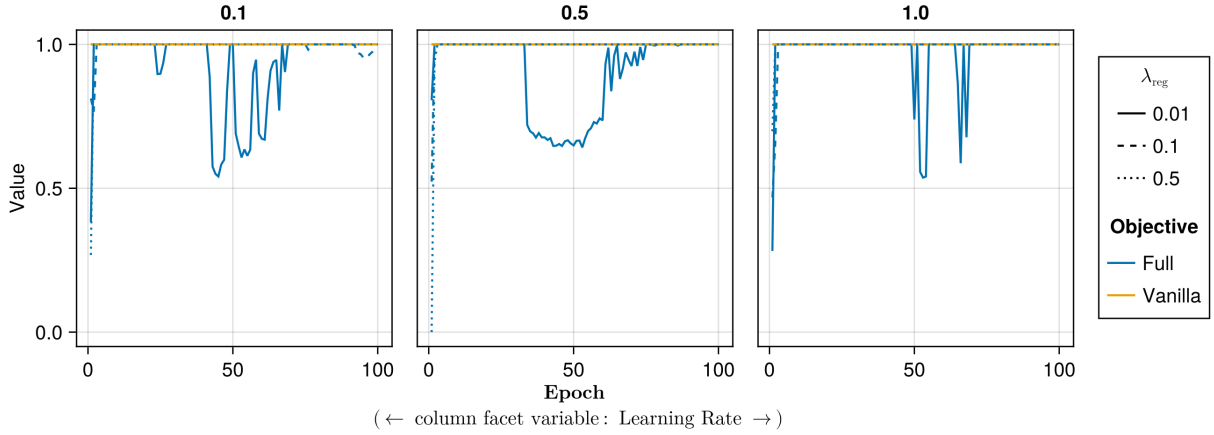

Figure 64: Proportion of mature counterfactuals in each epoch. Data: Moons.

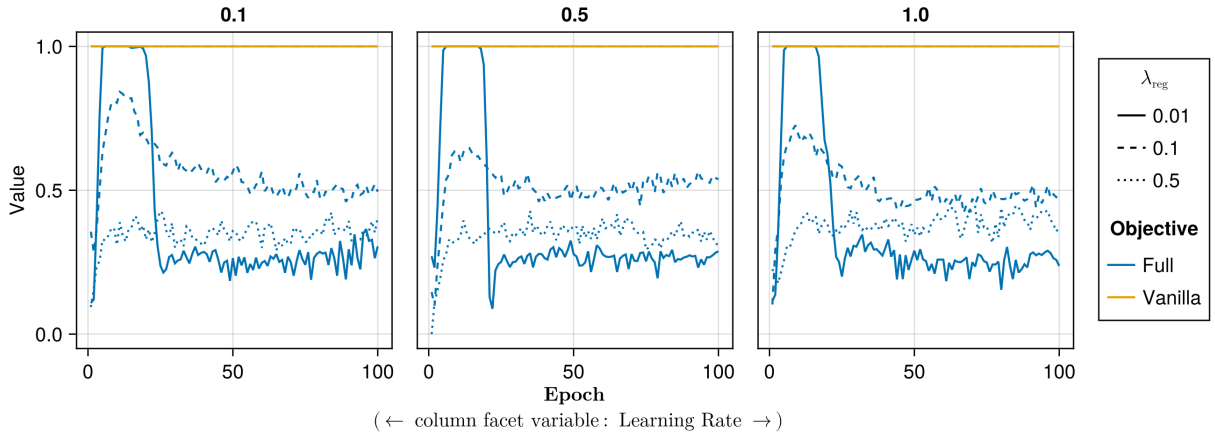

Figure 65: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

Appendix F Computation Details

F.1 Hardware

We performed our experiments on a high-performance cluster. Details about the cluster will be disclosed upon publication to avoid revealing information that might interfere with the double-blind review process. Since our experiments involve highly parallel tasks and rather small models by today's standard, we have relied on distributed computing across multiple central processing units (CPU). Graphical processing units (GPU) were not required.

F.1.1 Grid Searches

Model training for the largest grid searches with 270 unique parameter combinations was parallelized across 34 CPUs with 2GB memory each. The time to completion varied by dataset for reasons discussed in Section 5: 0h49m (*Moons*), 1h4m (*Linearly Separable*), 1h49m (*Circles*), 3h52m (*Overlapping*). Model evaluations for large grid searches were parallelized across 20 CPUs with 3GB memory each. Evaluations for all data sets took less than one hour (<1h) to complete.

F.1.2 Tuning

For tuning of selected hyperparameters, we distributed the task of generating counterfactuals during training across 40 CPUs with 2GB memory each for all tabular datasets. Except for the *Adult* dataset, all training runs were completed in less that half an hour (<0h30m). The *Adult* dataset took around 0h35m to complete. Evaluations across 20 CPUs with 3GB memory each generally took less than 0h30m to complete. For *MNIST*, we relied on 100 CPUs with 2GB memory each. For the *MLP*, training of all models could be completed in 1h30m, while the evaluation across 20 CPUs (6GB memory) took 4h12m. For the *CNN*, training of all models took ~8h, with conventionally trained models taking ~0h15m each and model with CT taking ~0h30m-0h45m each.

F.2 Software

All computations were performed in the Julia Programming Language (Bezanson et al. 2017). We have developed a package for counterfactual training that leverages and extends the functionality provided by several existing packages, most notably CounterfactualExplanations.jl (Altmeyer, Deursen, and Liem 2023) and the Flux.jl library for deep learning (Michael Innes et al. 2018; Mike Innes 2018). For data-wrangling and presentation-ready tables we relied on DataFrames.jl (Bouchet-Valat and Kamiski 2023) and PrettyTables.jl (Chagas et al. 2024), respectively. For plots and visualizations we used both Plots.jl (Christ et al. 2023) and Makie.jl (Danisch and Krumbiegel 2021), in particular AlgebraOfGraphics.jl. To distribute computational tasks across multiple processors, we have relied on MPI.jl (Byrne, Wilcox, and Churavy 2021).