

---

# 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](mailto: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

March 15, 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 (1) plausible with respect to the underlying data and (2) actionable with respect to the user-defined 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 ad-hoc 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 explanations while maintaining high predictive performance.

**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 **representation learning**: instead of relying on features and rules that are carefully hand-crafted by humans, modern machine learning (ML) models are tasked

18 with learning representations directly from data, guided by narrow objectives such as predictive accuracy (Goodfellow,  
 19 Bengio, and Courville 2016). Modern advances in computing have made it possible to provide such models with  
 20 ever-growing degrees of freedom to achieve that task, which frequently allows them to outperform traditionally more  
 21 parsimonious models. Unfortunately, in doing so, models learn increasingly complex and highly sensitive representa-  
 22 tions that humans can no longer easily interpret.  
 23 The trend towards complexity for the sake of performance has come under serious scrutiny in recent years. At the  
 24 very cusp of the deep learning (DL) revolution, Szegedy et al. (2014) showed that artificial neural networks (ANN)  
 25 are sensitive to adversarial examples (AEs): perturbed versions of data instances that yield vastly different model  
 26 predictions despite being “imperceptible” in that they are semantically indifferent from their factual counterparts.  
 27 Even though some partially effective mitigation strategies have been proposed—most notably **adversarial training**  
 28 (Goodfellow, Shlens, and Szegedy 2015)—truly robust deep learning remains unattainable even for models that are  
 29 considered “shallow” by today’s standards (Kolter 2023).  
 30 Part of the problem is that the high degrees of freedom provide room for many solutions that are locally optimal with  
 31 respect to narrow objectives (Wilson 2020).<sup>1</sup> Indeed, recent work on the so-called “lottery ticket hypothesis” suggests  
 32 that modern neural networks can be pruned by up to 90% while preserving their predictive performance (Frankle  
 33 and Carbin 2019). Similarly, Zhang et al. (2021) showed that state-of-the-art neural networks are expressive enough  
 34 to fit randomly labeled data. Thus, looking at the predictive performance alone, the solutions may seem to provide  
 35 compelling explanations for the data, when in fact they are based on purely associative, semantically meaningless  
 36 patterns. This poses two challenges. Firstly, there is no dependable way to verify if representations correspond to  
 37 meaningful, plausible explanations. Secondly, even if we could resolve the first challenge, it remains undecided how  
 38 to ensure that models can *only* learn valuable explanations.  
 39 The first challenge has attracted an abundance of research on **explainable AI** (XAI), a paradigm that focuses on the  
 40 development of tools to derive (post-hoc) explanations from complex model representations. Such explanations should  
 41 mitigate a scenario in which practitioners deploy opaque models and blindly rely on their predictions. On countless  
 42 occasions, this has happened in practice and caused real harms to people who were adversely and unfairly affected  
 43 by automated decision-making (ADM) systems involving opaque models (McGregor 2021; O’Neil 2016). Effective  
 44 XAI tools can aid us in monitoring models and providing recourse to individuals to turn negative outcomes (e.g.,  
 45 “loan application rejected”) into positive ones (e.g., “application accepted”). In line with this, our work builds upon  
 46 **counterfactual explanations** (CE) proposed by Wachter, Mittelstadt, and Russell (2017) as an effective approach to  
 47 achieve this goal. CEs prescribe minimal changes for factual inputs that, if implemented, would prompt some fitted  
 48 model to produce a desired output.  
 49 To our surprise, the second challenge has not yet attracted major research interest. Specifically, there has been no  
 50 concerted effort towards improving the “explanatory capacity” of models, i.e., the degree to which learned represen-  
 51 tations correspond to explanations that are **interpretable** and deemed **plausible** by humans (see Def. 3.1). Instead,  
 52 the choice has generally been to improve the ability of XAI tools to identify the subset of explanations that are both  
 53 plausible and valid for any given model, independent of whether the learned representations are in fact compatible  
 54 with plausible explanations (Altmeyer et al. 2024). Fortunately, recent findings indicate that improved explanatory  
 55 capacity can arise as a consequence of regularization techniques aimed at other training objectives such as generative  
 56 capacity, generalization, or robustness (Altmeyer et al. 2024; Augustin, Meinke, and Hein 2020; Schut et al. 2021).  
 57 As further discussed in Section 2, our work consolidates these findings within a single objective.  
 58 **Specifically, we introduce Counterfactual Training (CT):** a novel training regime explicitly geared towards improv-  
 59 ing the explainability of models. In high-level terms, we define this concept as as the extent to which valid explanations  
 60 derived for an opaque model are also deemed plausible with respect to the underlying data and the global actionability  
 61 constraints. To the best of our knowledge, our framework represents the first attempt to address this challenge by  
 62 employing counterfactual explanations already in the training phase.  
 63 The remainder of this manuscript is structured as follows. Section 2 presents related work, focusing on the link between  
 64 AEs and CEs. Then follow our two principal contributions. In Section 3, we introduce our methodological framework  
 65 and show theoretically that it can be employed to enforce global actionability constraints. In Section 4, through  
 66 extensive experiments, we demonstrate that CT substantially improves explainability without sacrificing predictive  
 67 performance. We discuss the challenges in Section 5 and conclude in Section 6 that CT is a promising approach  
 68 towards making opaque models more trustworthy.

---

<sup>1</sup>We follow the standard ML convention, where “degrees of freedom” refer to the number of parameters estimated from data.

## 69 2 Related Literature

70 To make the desiderata for our framework more concrete, we follow Augustin, Meinke, and Hein (2020) in tying the  
 71 concept of explainability to the quality of CEs that can be generated for a given model. The authors show that CEs—  
 72 understood as minimal input perturbations that yield some desired model prediction—tend to be more meaningful if the  
 73 underlying model is more robust to adversarial examples. We can make intuitive sense of this finding when looking  
 74 at adversarial training (AT) through the lens of representation learning with high degrees of freedom. As argued  
 75 before, learned representations may be sensitive to producing implausible explanations and mispredicting for worst-  
 76 case counterfactuals (i.e., AEs). Thus, by inducing models to “unlearn” susceptibility to such examples, adversarial  
 77 training can effectively remove implausible explanations from the solution space.

### 78 2.1 Adversarial Examples are Counterfactual Explanations

79 This interpretation of the link between explainability through counterfactuals on one side and robustness to adversarial  
 80 examples on the other is backed by empirical evidence. Sauer and Geiger (2021) demonstrates that using counter-  
 81 factual images during classifier training improves model robustness. Similarly, Abbasnejad et al. (2020) argue that  
 82 counterfactuals represent potentially useful training data in machine learning, especially in supervised settings where  
 83 inputs may be reasonably mapped to multiple outputs. They, too, demonstrate that augmenting the training data of  
 84 image classifiers can improve generalization. Finally, Teney, Abbasnejad, and Hengel (2020) propose an approach  
 85 using counterfactuals in training that does not rely on data augmentation: they argue that counterfactual pairs typically  
 86 already exist in training datasets. Specifically, their approach relies on identifying similar input samples with different  
 87 annotations and ensuring that the gradient of the classifier aligns with the vector between such pairs of counterfactual  
 88 inputs using the cosine distance as the loss function.

89 In the natural language processing (NLP) domain, CEs have also been used to improve models through data augmen-  
 90 tation. Wu et al. (2021) proposes *Polyjuice*, a general-purpose counterfactual generator for language models. The  
 91 authors empirically demonstrate that the augmentation of training data with their method improves robustness in a  
 92 number of NLP tasks. Balashankar et al. (2023) similarly uses *Polyjuice* to augment NLP datasets through diverse  
 93 counterfactuals and show that classifier robustness improves by up to 20%. Finally, Luu and Inoue (2023) introduces  
 94 Counterfactual Adversarial Training (CAT) that also aims to improve generalization and robustness of language mod-  
 95 els through a three-step procedure: the authors identify training samples that are subject to high predictive uncertainty,  
 96 generate CEs for them, and fine-tune the language model on a dataset augmented with the CEs.

97 There have also been several attempts at formalizing the relationship between counterfactual explanations and adver-  
 98 sarial examples. Pointing to clear similarities in how CEs and AEs are generated, Freiesleben (2022) makes the case  
 99 for jointly studying the opaqueness and robustness problems in representation learning. Formally, AEs can be seen as  
 100 the subset of CEs for which misclassification is achieved (Freiesleben 2022). Similarly, Pawelczyk et al. (2022) shows  
 101 that CEs and AEs are equivalent under certain conditions.

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

### 113 2.2 Aligning Representations with Plausible Explanations

114 Improving the adversarial robustness of models is not the only path towards aligning representations with plausible  
 115 explanations. In a work closely related to this one, Altmeyer et al. (2024) shows that explainability can be improved  
 116 through model averaging and refined model objectives. The authors propose a way to generate counterfactuals that  
 117 are maximally faithful to the model in that they are consistent with what the model has learned about the underlying  
 118 data. Formally, they rely on tools from energy-based modelling (Teh et al. 2003) to minimize the divergence between  
 119 the distribution of counterfactuals and the conditional posterior over inputs learned by the model. Their proposed  
 120 counterfactual explainer, *ECCCo*, yields plausible explanations if and only if the underlying model has learned repre-  
 121 sentations that align with them. The authors find that both deep ensembles (Lakshminarayanan, Pritzel, and Blundell  
 122 2017) and joint energy-based models (JEMs) (Grathwohl et al. 2020) tend to do well in this regard.

123 Once again it helps to look at these findings through the lens of representation learning with high degrees of freedom.  
 124 Deep ensembles are approximate Bayesian model averages, which are most called for when models are underspecified  
 125 by the available data (Wilson 2020). Averaging across solutions mitigates the aforementioned risk of relying on a  
 126 single locally optimal representations that corresponds to semantically meaningless explanations for the data. Previous  
 127 work of Schut et al. (2021) similarly found that generating plausible (“interpretable”) CEs is almost trivial for deep  
 128 ensembles that have also undergone adversarial training. The case for JEMs is even clearer: they involve a hybrid  
 129 objective that induces both high predictive performance and generative capacity (Grathwohl et al. 2020). This is  
 130 closely related to the idea of aligning models with plausible explanations and has inspired our CT objective.

### 131 3 Counterfactual Training

132 In this section we propose our novel counterfactual training objective. In CT, we combine ideas from adversarial  
 133 training, counterfactual explanations, and energy-based modelling with the explicit goal of aligning representations  
 134 with plausible explanations that comply with user-defined actionability constraints.

135 In the context of counterfactual explanations, plausibility has broadly been defined as the degree to which counter-  
 136 factuals comply with the underlying data-generating process (Altmeyer et al. 2024; Guidotti 2022; Poyiadzi et al.  
 137 2020). Plausibility is a necessary but insufficient condition for using CEs to provide algorithmic recourse (AR) to  
 138 individuals (negatively) affected by opaque models. To be actionable, AR recommendations must also be attainable.  
 139 A plausible CE for a rejected 20-year-old loan applicant, for example, might reveal that their application would have  
 140 been accepted, if only they were 20 years older. Ignoring all other features, this would comply with the definition of  
 141 plausibility if 40-year-old individuals were in fact more credit-worthy on average than young adults. But of course  
 142 this CE does not qualify for providing actionable recourse to the applicant since *age* is not a (directly) mutable feature.  
 143 Counterfactual training aims to improve model explainability by aligning models with counterfactuals that meet both  
 144 desiderata: plausibility and actionability. Formally, we define explainability as follows:

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

149 We say that  $M_\theta$  is **explainable** to the extent that faithfully generated counterfactuals are plausible and actionable. We  
 150 define these properties as follows:

- 151 1. (Plausibility)  $\int^A p(\mathbf{x}' | \mathbf{y}^+) d\mathbf{x} \rightarrow 1$  where  $A$  is some small region around  $\mathbf{x}'$ .
- 152 2. (Actionability) Permutations  $\Delta$  are subject to some actionability constraints.
- 153 3. (Faithfulness)  $\int^A p_\theta(\mathbf{x}' | \mathbf{y}^+) d\mathbf{x} \rightarrow 1$  where  $A$  is defined as above.

154 where  $p_\theta(\mathbf{x} | \mathbf{y}^+)$  denotes the conditional posterior over inputs.

155 The characterization of faithfulness and plausibility in Def. 3.1 is the same as in Altmeyer et al. (2024), with adapted  
 156 notation. Intuitively, plausible counterfactuals are consistent with the data and faithful counterfactuals are consistent  
 157 with what the model has learned about input data. Actionability constraints in Def. 3.1 vary and depend on the context  
 158 in which  $M_\theta$  is deployed. In this work, we focus on domain and mutability constraints for individual features  $x_d$  for  
 159  $d = 1, \dots, D$ . We limit ourselves to classification tasks for reasons discussed in Section 5.

#### 160 3.1 Our Proposed Objective

161 Let  $\mathbf{x}'_t$  for  $t = 0, \dots, T$  denote a counterfactual explanation generated through gradient descent over  $T$  iterations  
 162 as initially proposed by Wachter, Mittelstadt, and Russell (2017). For our purposes, we let  $T$  vary and consider the  
 163 counterfactual search as converged as soon as the predicted probability for the target class has reached a pre-determined  
 164 threshold,  $\tau : \mathcal{S}(M_\theta(\mathbf{x}'))[y^+] \geq \tau$ , where  $\mathcal{S}$  is the softmax function.<sup>2</sup>  
 165 To train models with high explainability as defined in Def. 3.1, we propose to leverage counterfactuals in the following  
 166 objective:

$$\begin{aligned} \min_{\theta} & \text{yloss}(M_\theta(\mathbf{x}), \mathbf{y}) + \lambda_{\text{div}} \text{div}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) + \lambda_{\text{adv}} \text{advloss}(M_\theta(\mathbf{x}'_{t \leq T}), \mathbf{y}) \\ & + \lambda_{\text{reg}} \text{ridge}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) \end{aligned} \quad (1)$$

---

<sup>2</sup>For detailed background information on gradient-based counterfactual search and convergence see supplementary appendix.

167 where  $y\text{loss}(\cdot)$  is a classification loss that induces discriminative performance (e.g., cross-entropy). The second and  
 168 third terms are explained in detail below. For now, they can be summarized as inducing explainability directly and  
 169 indirectly by penalizing: (1) the contrastive divergence,  $\text{div}(\cdot)$ , between mature counterfactuals  $\mathbf{x}'_T$  and observed  
 170 samples  $\mathbf{x}^+ \in \mathcal{X}^+ = \{\mathbf{x} : y = y^+\}$  in the target class  $y^+$ , and, (2) the adversarial loss,  $\text{advloss}(\cdot)$ , with respect to  
 171 nascent counterfactuals  $\mathbf{x}'_{t \leq T}$ . Finally,  $\text{ridge}(\cdot)$  denotes a Ridge penalty ( $\ell_2$ -norm) that regularizes the magnitude of  
 172 the energy terms involved in  $\text{div}(\cdot)$  (Du and Mordatch 2020). The trade-off between the components can be governed  
 173 through penalties  $\lambda_{\text{div}}$ ,  $\lambda_{\text{adv}}$  and  $\lambda_{\text{reg}}$ .

### 174 3.2 Directly Inducing Explainability with Contrastive Divergence

175 As observed by Grathwohl et al. (2020), any classifier can be re-interpreted as a joint energy-based model (JEM)  
 176 that learns to discriminate output classes conditional on the observed (training) samples from  $p(\mathbf{x})$  and the generated  
 177 samples from  $p_\theta(\mathbf{x})$ . The authors show that JEMs can be trained to perform well at both tasks by directly maximizing  
 178 the joint log-likelihood factorized as  $\log p_\theta(\mathbf{x}, \mathbf{y}) = \log p_\theta(\mathbf{y}|\mathbf{x}) + \log p_\theta(\mathbf{x})$ . The first term can be optimized using  
 179 conventional cross-entropy as in Equation 1. To optimize  $\log p_\theta(\mathbf{x})$ , Grathwohl et al. (2020) minimizes the contrastive  
 180 divergence between the observed samples from  $p(\mathbf{x})$  and generated samples from  $p_\theta(\mathbf{x})$ .

181 A key empirical finding in Altmeyer et al. (2024) was that JEMs tend to do well with respect to the plausibility  
 182 objective in Def. 3.1. This follows directly if we consider samples drawn from  $p_\theta(\mathbf{x})$  as counterfactuals because  
 183 the JEM objective effectively minimizes the divergence between the conditional posterior and  $p(\mathbf{x}|y^+)$ . To generate  
 184 samples, Grathwohl et al. (2020) relies on Stochastic Gradient Langevin Dynamics (SGLD) using an uninformative  
 185 prior for initialization but we depart from their methodology. Instead of SGLD, we propose to use counterfactual  
 186 explainers to generate counterfactuals of observed training samples. Specifically, we have:

$$\text{div}(\mathbf{x}^+, \mathbf{x}'_T, y; \theta) = \mathcal{E}_\theta(\mathbf{x}^+, y) - \mathcal{E}_\theta(\mathbf{x}'_T, y) \quad (2)$$

187 where  $\mathcal{E}_\theta(\cdot)$  denotes the energy function defined as  $\mathcal{E}_\theta(\mathbf{x}, y) = -\mathbf{M}_\theta(\mathbf{x})[y^+]$  where  $y^+$  denotes the index of the  
 188 randomly drawn target class,  $y^+ \sim p(y)$ . Conditional on the target class  $y^+$ ,  $\mathbf{x}'_T$  denotes a mature counterfactual for a  
 189 randomly sampled factual from a non-target class generated with a gradient-based CE generator for up to  $T$  iterations.  
 190 Mature counterfactuals are ones that have either reached convergence wrt. the decision threshold  $\tau$  or exhausted  $T$ .

191 Intuitively, the gradient of Equation 2 decreases the energy of observed training samples (positive samples) while  
 192 increasing the energy of counterfactuals (negative samples) (Du and Mordatch 2020). As the counterfactuals get more  
 193 plausible (Def. 3.1) during training, these opposing effects gradually balance each other out (Lippe 2024).

194 The departure from SGLD of (Grathwohl et al. 2020) allows us to tap into the vast repertoire of explainers that have  
 195 been proposed in the literature to meet different desiderata. For example, many methods facilitate the imposition of  
 196 domain and mutability constraints. In principle, any existing approach for generating counterfactual explanations is  
 197 viable, so long as it does not violate the faithfulness condition. Like JEMs (Murphy 2022), CT can be considered a  
 198 form of contrastive representation learning.

### 199 3.3 Indirectly Inducing Explainability with Adversarial Robustness

200 Based on our analysis in Section 2, counterfactuals  $\mathbf{x}'$  can be repurposed as additional training samples (Balashankar  
 201 et al. 2023; Luu and Inoue 2023) or adversarial examples (Freiesleben 2022; Pawelczyk et al. 2022). This leaves some  
 202 flexibility wrt. the choice for  $\text{advloss}(\cdot)$  in Equation 1. An intuitive functional form, but likely not the only sensible  
 203 choice, is inspired by adversarial training:

$$\begin{aligned} \text{advloss}(\mathbf{M}_\theta(\mathbf{x}'_{t \leq T}), \mathbf{y}; \varepsilon) &= \text{yloss}(\mathbf{M}_\theta(\mathbf{x}'_{t_\varepsilon}), \mathbf{y}) \\ t_\varepsilon &= \max_t \{t : \|\Delta_t\|_\infty < \varepsilon\} \end{aligned} \quad (3)$$

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

### 210 3.4 Encoding Actionability Constraints

211 Many existing counterfactual explainers support domain and mutability constraints out-of-the-box. In fact, both types  
 212 of constraints can be implemented for any counterfactual explainer that relies on gradient descent in the feature space  
 213 for optimization (Altmeyer, Deursen, and Liem 2023). In this context, domain constraints can be imposed by simply

214 projecting counterfactuals back to the specified domain, if the previous gradient step resulted in updated feature values  
 215 that were out-of-domain. Mutability constraints can similarly be enforced by setting partial derivatives to zero to  
 216 ensure that features are only perturbed in the allowed direction, if at all.

217 Since such actionability constraints are binding at test time, we should also impose them when generating  $\mathbf{x}'$  during  
 218 each training iteration to inform model representations. Through their effect on  $\mathbf{x}'$ , both types of constraints influence  
 219 model outcomes via Equation 2. Here it is crucial that we avoid penalizing implausibility that arises due to mutability  
 220 constraints. For any mutability-constrained feature  $d$  this can be achieved by enforcing  $\mathbf{x}^+[d] - \mathbf{x}'[d] := 0$  whenever  
 221 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:

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

225 From a Bayesian perspective, setting  $\mathbf{x}^+[d] := \mathbf{x}'[d]$  can be understood as assuming a point mass prior for  $p(\mathbf{x}^+)$   
 226 with respect to feature  $d$ . Intuitively, we think of this simply in terms ignoring implausibility costs with respect  
 227 to immutable features, which effectively forces the model to instead seek plausibility with respect to the remaining  
 228 features. This in turn results in lower overall sensitivity to immutable features, which we demonstrate empirically for  
 229 different classifiers in Section 4. Under certain conditions, this result holds theoretically:<sup>3</sup>

230 **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  
 231 softmax activation  $\mathcal{S}$  where  $y \in \{1, \dots, K\} = \mathcal{K}$  and  $\mathbf{x} \in \mathbb{R}^D$ . If we assume multivariate Gaussian class densities with  
 232 common diagonal covariance matrix  $\Sigma_k = \Sigma$  for all  $k \in \mathcal{K}$ , then protecting an immutable feature from the contrastive  
 233 divergence penalty will result in lower classifier sensitivity to that feature relative to the remaining features, provided  
 234 that at least one of those is discriminative and mutable.*

235 It is worth highlighting that Proposition 3.1 assumes independence of features. This raises a valid concern about  
 236 the effect of protecting immutable features in the presence of proxies that remain unprotected. We address this in  
 237 Section 5.

### 238 3.5 Example (Prediction of Consumer Credit Default)

239 Suppose we are interested in predicting the likelihood that loan applicants default on their credit. We have access to  
 240 historical data on previous loan takers comprised of a binary outcome variable ( $y \in \{1 = \text{default}, 2 = \text{no default}\}$ )  
 241 with two input features: (1) the subjects' *age*, which we define as immutable, and (2) the subjects' existing level of  
 242 *debt*, which we define as mutable.

243 We have simulated this scenario using synthetic data with independent *age* and *debt*, and Gaussian class-conditional  
 244 densities in Figure 1. The four panels show the outcomes for different training procedures using the same model  
 245 architectures (a linear classifier). In panels (a) and (c) we have trained the models conventionally, while in panels (b)  
 246 and (d) we used CT.

247 In all cases, all counterfactuals (stars) are valid—they have cross the decision boundary (green)—but their quality  
 248 differs. In panel (a), they are not plausible: they do not comply with the distribution of the factuals in  $y^+$  to the point  
 249 where they form a clearly distinguishable cluster. In panel (b), they are highly plausible, meeting the first objective  
 250 of Def. 3.1. In panel (c), the CEs involve substantial reductions in *debt* for younger applicants. By comparison,  
 251 counterfactual paths are shorter on average in panel (d) where we have protected the immutable *age* as described in  
 252 Section 3.4. Due to the classifier's lower sensitivity to *age*, recommendations with respect to *debt* are much more  
 253 homogenous and do not unfairly punish younger individuals. These counterfactuals are also plausible with respect to  
 254 the mutable feature. Thus, we consider the model in panel (d) as the most explainable according to Def. 3.1.

## 255 4 Experiments

256 In our experiments we seek to answer the following three research questions:

- 257    1. To what extent does the counterfactual training objective as it is defined in Equation 1 induce models to learn  
 258      plausible explanations?  
 259    2. To what extent does the CT objective produce more favorable algorithmic recourse outcomes in the presence  
 260      of actionability constraints?  
 261    3. What are the effects of hyperparameter selection wrt. the CT objective?

---

<sup>3</sup>For the proof, see the supplementary appendix.

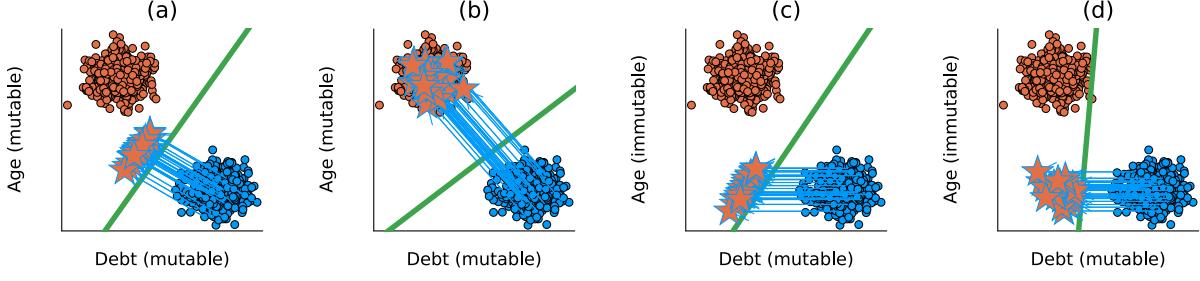


Figure 1: Illustration of how CT improves model explainability: (a) conventional training, all mutable; (b) CT, all mutable; (c) conventional, *age* immutable; (d) CT, *age* immutable. The decision boundary is shown in green along with training data colored according to their ground-truth label:  $y^+ = 2$  (orange) and  $y^- = 1$  (blue). Stars indicate CEs for the target class.

#### 262 4.1 Experimental Setup

263 Our key outcome of interest is improved explainability (Def. 3.1). To this end, we focus primarily on the plausibility  
 264 and cost of faithfully generated counterfactuals at test time. To measure the cost of counterfactuals, we follow the  
 265 standard convention of using distances ( $\ell_1$ -norm) between factuals and counterfactuals as a proxy. For plausibility,  
 266 we assess how similar counterfactuals are to observed samples in the target domain,  $\mathbf{X}' \subset \mathcal{X}^+$ . We rely on the  
 267 distance-based metric used in Altmeyer et al. (2024),

$$\text{IP}(\mathbf{x}', \mathbf{X}^+) = \frac{1}{|\mathbf{X}^+|} \sum_{\mathbf{x} \in \mathbf{X}^+} \text{dist}(\mathbf{x}', \mathbf{x}) \quad (4)$$

268 and introduce a novel divergence metric,

$$\text{IP}^*(\mathbf{X}', \mathbf{X}^+) = \text{MMD}(\mathbf{X}', \mathbf{X}^+) \quad (5)$$

269 where  $\mathbf{X}'$  denotes a collection of counterfactuals and  $\text{MMD}(\cdot)$  is an unbiased estimate of the squared population  
 270 maximum mean discrepancy (Gretton et al. 2012). The metric in Equation 5 is equal to zero iff the two distributions  
 271 are the same,  $\mathbf{X}' = \mathbf{X}^+$ .

272 In addition to cost and plausibility, we compute other standard metrics to evaluate counterfactuals including validity  
 273 and redundancy. Finally, we also assess the predictive performance of models using standard metrics, including robust  
 274 accuracy estimated on adversarially perturbed data using FGSM (Goodfellow, Shlens, and Szegedy 2015).

275 We run the experiments with three gradient-based generators: *Generic* of Wachter, Mittelstadt, and Russell (2017) as  
 276 a simple baseline approach, *REVISE* (Joshi et al. 2019) that aims to generate plausible counterfactuals using a surro-  
 277 gate Variational Autoencoder (VAE), and *ECCo*—the generator of Altmeyer et al. (2024) but without the conformal  
 278 prediction component—as a method that directly targets both faithfulness and plausibility of the counterfactuals.

279 We make use of nine classification datasets common in the CE/AR literature. Four of them are synthetic with two  
 280 classes and different characteristics: linearly separable clusters (*LS*), overlapping clusters (*OL*), concentric circles  
 281 (*Circ*), and interlocking moons (*Moon*). These datasets are generated using the library of (Altmeyer, Deursen, and  
 282 Liem 2023) and we present them in the supplementary appendix. Next, we have four real-world binary tabular datasets  
 283 from the domain of economics: *Adult* (a.k.a. Census data) of (Becker and Kohavi 1996), California housing (*CH*) of  
 284 (Pace and Barry 1997), Default of Credit Card Clients (*Cred*) of (Yeh 2016), and Give Me Some Credit (*GMSC*) of  
 285 (Kaggle 2011). Finally, for the convenience of illustration, we use of the 10-class *MNIST* vision dataset (LeCun 1998).

286 To assess our proposed training regime, we investigate the improvements in performance metrics when using CT on  
 287 top of a weak baseline (BL): specifically, a multilayer perceptron (*MLP*). This is the best way to get a clear picture of  
 288 how effective CT is and consistent with how assessment is done in the related literature (Ross, Lakkaraju, and Bastani  
 289 2024; Teney, Abbasnejad, and Hengel 2020; Goodfellow, Shlens, and Szegedy 2015).

#### 290 4.2 Experimental Results

##### 291 4.2.1 Plausibility

292 Table 1 presents our main empirical findings. For all datasets except *OL* and across all test settings, the average  
 293 distance of CEs from observed samples in the target class is reduced, indicating improved plausibility. The magnitude  
 294 of improvements varies. For the simple synthetic datasets, distance reductions range from around 20-40% (*LS*, *Moon*)

Table 1: Key performance metrics across all datasets. **Plausibility**: Columns 2-6 show the percentage reduction in implausibility (IP) for varying degrees of the energy penalty used for *ECCo* at test time; column 7 shows the reduction in IP\* (MMD), aggregated across all test specifications. **Accuracy** (columns 8-11): test accuracies and robust accuracies (Acc\*) for CT and the baseline (BL). **Actionability** (column 9): average reduction in costs when imposing mutability constraints.

<b>Data</b>	<b>IP</b>	<b>IP</b>	<b>IP</b>	<b>IP</b>	<b>IP</b>	<b>IP*</b>	<b>Acc.</b>	<b>Acc.</b>	<b>Acc.*</b>	<b>Acc.*</b>	<b>Cost</b>
	(-%) $\lambda_{\text{egy}}$ 0.1	(-%) 0.5	(-%) 1.0	(-%) 5.0	(-%) 10.0	(agg.)	(CT)	(BL)	(CT)	(BL)	(-%)
Adult	2.9	3.4	3.5	2.9	3.2	34.8	0.85	0.85	0.83	0.41	
CH	9.6	9.3	10.4	11.9	14.6	66.6	0.79	0.85	0.76	0.75	
Circ	56.5	57.1	56.5	58.5	49.3	93.4	1.0	1.0	0.99	1.0	35.0
Cred	6.7	6.2	7.2	7.0	7.8	51.6	0.71	0.71	0.7	0.52	
GMSC	11.0	13.4	13.4	21.4	27.9	77.9	0.61	0.75	0.58	0.42	
LS	27.1	26.7	26.6	27.1	38.6	54.5	1.0	1.0	1.0	1.0	26.3
MNIST	9.1	8.3	8.1	6.1	3.5	-2.3	0.9	0.92	0.84	0.78	
Moon	20.4	21.4	21.6	19.0	19.8	27.6	1.0	1.0	1.0	1.0	23.4
OL	-6.7	-6.2	-6.1	-2.8	-1.4	-25.5	0.92	0.91	0.91	0.91	15.5

295 to almost 60% (*Circ*). For the real-world tabular datasets, improvements tend to be smaller but still substantial, with  
 296 around 10-15% for *CH*, 11-28% for *GMSC*, 7-8% for *Cred*, and around 3% for *Adult*. For our only vision dataset  
 297 (*MNIST*), distances are reduced by up to 9%. The results for our proposed divergence metric are qualitatively similar,  
 298 but generally even more pronounced: for the *Circ* dataset, implausibility is reduced by almost 94% to virtually zero  
 299 as we verified by the absolute outcome. Improvements for other datasets range from 28% (*Moon*) to 78% (*GMSC*).  
 300 For *OL* the reduction is negative, consistent with the distance-based metric. *MNIST* is the only dataset for which the  
 301 distance and divergence metrics disagree. Upon visual inspection of the image counterfactuals we find that CT clearly  
 302 improves plausibility (see appendix).

#### 303 4.2.2 Predictive Performance

304 Test accuracy for CT is virtually identical to the baseline for *Adult*, *Circ*, *LS*, *Moon*, and *OL*, and even slightly improved  
 305 for *Cred*. Exceptions to this general pattern are *MNIST*, *CH*, and *GMSC*, for which we observe a reduction in test  
 306 accuracy of 2, 5, and 15 percentage points respectively. When looking at robust test accuracies (Acc\*) for these  
 307 datasets in particular, we find that CT strongly outperforms the baseline. In fact, we find that CT improves adversarial  
 308 robustness on all datasets.

#### 309 4.2.3 Actionability

310 In Section 3, we show that our proposed way for encoding mutability constraints leads to lower classifier sensitivity  
 311 wrt. immutable features for linear models, tilting the decision boundary in favour of mutable features instead. For  
 312 binding constraints at test time, this leads to shorter counterfactual paths and hence smaller average costs ( $\ell_1$ -norm) to  
 313 individuals. To extend this to the non-linear case, we test the effect of imposing mutability constraints empirically for  
 314 our synthetic data using the same evaluation scheme as above. The final row in Table 1 reports the average reduction in  
 315 costs for CT compared to the “vanilla” baseline, when imposing that either the first or the second feature is immutable.  
 316 In all cases, costs are reduced substantially, indicating that classifiers trained with CT are indeed more sensitive to  
 317 mutable features.

#### 318 4.2.4 Impact of hyperparameter settings.

319 We test the impact of three types of hyperparameters; our complete results are in the supplementary appendix.  
 320 We note that CT is highly sensitive to the choice of a CE generator and its hyperparameters but (a) there are manageable  
 321 patterns and (b) we can typically identify settings that improve either plausibility or cost, and commonly both of them  
 322 at the same time. For example, *REVISE* tends to perform the worst, most likely because it uses a surrogate VAE to  
 323 generate counterfactuals which impedes faithfulness (Altmeyer et al. 2024). Increasing  $T$ , the maximum number of  
 324 steps, generally yields better outcomes because more CEs can mature in each training epoch. The impact of  $\tau$ , the  
 325 required decision threshold is more difficult to predict. On “harder” datasets it may be difficult to satisfy high  $\tau$  for any  
 326 given sample (i.e., also factuals) and so increasing this threshold does not seem to correlate with better outcomes. In  
 327 fact, the choice of  $\tau = 0.5$  generally leads to optimal results because it is associated with high proportions of mature  
 328 counterfactuals.

329 The strength of the energy regularization,  $\lambda_{\text{reg}}$  is highly impactful and leads to poor performance in terms of decreased  
 330 plausibility and increased costs if insufficiently high. The sensitivity with respect to  $\lambda_{\text{div}}$  and  $\lambda_{\text{adv}}$  is much less evident.

- 331 While high values of  $\lambda_{\text{reg}}$  may increase the variability in outcomes when combined with high values of  $\lambda_{\text{div}}$  or  $\lambda_{\text{adv}}$ ,  
 332 this effect is not very pronounced.  
 333 The effectiveness and stability of CT is positively associated with the number of counterfactuals generated during  
 334 each training epoch. We also confirm that a higher number of training epochs is beneficial. Interestingly, we observed  
 335 desired improvements when CT was combined with conventional training and applied only for the final 50% of epochs  
 336 of the complete training process. Put differently, CT can improve the explainability of models in a fine-tuning manner.

## 337 5 Discussion

338 As our results indicate, counterfactual training produces models that are more explainable. Nonetheless, it brings  
 339 about three important limitations.

340 *CT increases the training time of models.* CT can be more time-consuming than conventional training regimes. While  
 341 higher numbers of CEs per iteration positively impact the quality of solutions, they also increase the amount of com-  
 342 putations. Relatively small grids with 270 settings can take almost four hours for more demanding datasets on a  
 343 high-performance computing cluster with 34 2GB CPUs.<sup>4</sup> Three factors attenuate this effect. First, CT amortizes the  
 344 cost of CEs for the training samples. Second, we find that it can retain its value when used as a “fine-tuning” technique  
 345 for conventionally-trained models. Third, it yields itself to parallel execution, which we have leveraged for our own  
 346 experiments.

347 *Immutable features may have proxies.* We propose an approach to protect immutable features and thus increase the  
 348 actionability of the generated CEs. However, it requires that model owners define the mutability constraints for (all)  
 349 features considered by the model. Even if all immutable features are protected, there may exist proxies that are muta-  
 350 ble (and hence should not be protected) but preserve enough information about the principals to hinder the protections.  
 351 Delineating actionability is a major undecided challenge in the AR literature (see, e.g., ([Venkatasubramanian and Al-](#)  
 352 [fano 2020](#))) impacting the capacity of CT to fulfill its intended goal.

353 *Interventions on features may impact fairness.* We provide a tool that allows practitioners to modify the sensitivity of  
 354 a model with respect to certain features, which may have implication for the fair and equitable treatment of decision  
 355 subjects. As protecting a set of features leads the model to assign higher relative importance to unprotected features,  
 356 model owners could misuse our solution by enforcing explanations based on features that are more difficult to modify  
 357 by some (group of) individuals. For example, consider the Adult dataset used in our experiments, where *workclass*  
 358 or *education* may be more difficult to change for underprivileged groups. When applied irresponsibly, CT could  
 359 result in an unfairly assigned burden of recourse ([Sharma, Henderson, and Ghosh 2020](#)), threatening the equality of  
 360 opportunity in the system ([Bell et al. 2024](#)). Nonetheless, these phenomena are not specific to CT.

361 We also highlight several important directions for future research. Firstly, it is an interesting challenge to extend CT  
 362 beyond classification settings. Our formulation relies on the distinction between non-target class(es)  $y^-$  and target  
 363 class(es)  $y^+$  to generate counterfactuals through Equation 1. While  $y^-$  and  $y^+$  can be arbitrarily defined, CT requires  
 364 the output space  $\mathcal{Y}$  to be discrete. Thus, it does not apply to ML tasks where the change in outcome cannot be readily  
 365 quantified. Focus on classification models is a common restriction in research on CEs and AR. Other settings have  
 366 attracted some interest (e.g., regression in ([Spooner et al. 2021](#))), but there is little consensus how to robustly extend  
 367 the notion of CEs.

368 Secondly, our approach is susceptible to training instabilities. This problem has been recognized for JEMs ([Grathwohl](#)  
 369 [et al. 2020](#)) and even though we depart from the SGLD-based sampling, we still encounter considerable variability  
 370 in the outcomes. CT is exposed to two potential sources of instabilities: (1) the energy-based contrastive divergence  
 371 term in Equation 2, and (2) the underlying counterfactual explainers. We find several promising ways to mitigate this  
 372 problem: regularizing energy ( $\lambda_{\text{reg}}$ ), generating sufficiently many counterfactuals during each epoch, and including  
 373 only mature counterfactuals for contrastive divergence.

374 Finally, we believe that it is possible to substantially improve hyperparameter selection procedures. Our method  
 375 benefits from the tuning of certain key hyperparameters (see Section 4.2.4). In this work, we have relied exclusively  
 376 on grid search for this task. Future work on CT could benefit from investigating more sophisticated approaches.  
 377 Notably, CT is iterative which makes methods such as Bayesian or gradient-based optimization applicable (see, e.g.,  
 378 ([Bischl et al. 2023](#))).

## 379 6 Conclusion

380 State-of-the-art machine learning models are prone to learning complex representations that cannot be interpreted by  
 381 humans and existing post-hoc explainability approaches cannot guarantee that the explanations agree with the model’s  
 382 learned representation of data. As a step towards addressing this challenge, we introduced counterfactual training, a  
 383 novel training regime that incentivizes highly-explainable models. Our approach leads to explanations that are both  
 384 plausible—compliant with the underlying data-generating process—and actionable—compliant with user-specified

---

<sup>4</sup>See supplementary appendix for computational details.

385 mutability constraints—and thus meaningful to their recipients. Through extensive experiments we demonstrate that  
 386 CT satisfies its objectives while preserving the predictive performance of the models. Our approach can also be used  
 387 to fine-tune conventionally-trained models and achieve similar gains in explainability. Finally, this work showcases  
 388 that it is practical to improve models *and* their explanations at the same time.

## 389 References

- 390 Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. “Counterfactual  
 391 Vision and Language Learning.” In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition  
 392 (CVPR)*, 10041–51. <https://doi.org/10.1109/CVPR42600.2020.01006>.
- 393 Altmeyer, Patrick, Arie van Deursen, and Cynthia C. S. Liem. 2023. “Explaining Black-Box Models through Coun-  
 394 terfactuals.” In *Proceedings of the JuliaCon Conferences*, 1:130.
- 395 Altmeyer, Patrick, Mojtaba Farmanbar, Arie van Deursen, and Cynthia C. S. Liem. 2024. “Faithful Model Ex-  
 396 planations through Energy-Constrained Conformal Counterfactuals.” In *Proceedings of the Thirty-Eighth AAAI  
 397 Conference on Artificial Intelligence*, 38:10829–37. 10. <https://doi.org/10.1609/aaai.v38i10.28956>.
- 398 Augustin, Maximilian, Alexander Meinke, and Matthias Hein. 2020. “Adversarial Robustness on In- and Out-  
 399 Distribution Improves Explainability.” In *Computer Vision – ECCV 2020*, edited by Andrea Vedaldi, Horst Bischof,  
 400 Thomas Brox, and Jan-Michael Frahm, 228–45. Cham: Springer.
- 401 Balashankar, Ananth, Xuezhi Wang, Yao Qin, Ben Packer, Nithum Thain, Ed Chi, Jilin Chen, and Alex Beutel. 2023.  
 402 “Improving Classifier Robustness through Active Generative Counterfactual Data Augmentation.” In *Findings of  
 403 the Association for Computational Linguistics: EMNLP 2023*, 127–39. ACL. <https://doi.org/10.18653/v1/2023.f>  
 404 indings-emnlp.10.
- 405 Becker, Barry, and Ronny Kohavi. 1996. “Adult.” UCI Machine Learning Repository.
- 406 Bell, Andrew, Joao Fonseca, Carlo Abate, Francesco Bonchi, and Julia Stoyanovich. 2024. “Fairness in Algorithmic  
 407 Recourse Through the Lens of Substantive Equality of Opportunity.” <https://arxiv.org/abs/2401.16088>.
- 408 Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. “Julia: A Fresh Approach to Numerical  
 409 Computing.” *SIAM Review* 59 (1): 65–98. <https://doi.org/10.1137/141000671>.
- 410 Bischl, Bernd, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, et al. 2023.  
 411 “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges.” *WIREs Data Min-  
 412 ing and Knowledge Discovery* 13 (2): e1484. <https://doi.org/10.1002/widm.1484>.
- 413 Bouchet-Valat, Milan, and Bogumi Kamiski. 2023. “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” *Journal  
 414 of Statistical Software* 107 (4): 1–32. <https://doi.org/10.18637/jss.v107.i04>.
- 415 Byrne, Simon, Lucas C. Wilcox, and Valentin Churavy. 2021. “MPI.jl: Julia Bindings for the Message Passing  
 416 Interface.” *Proceedings of the JuliaCon Conferences* 1 (1): 68. <https://doi.org/10.21105/jcon.00068>.
- 417 Chagas, Ronan Arraes Jardim, Ben Baumgold, Glen Hertz, Hendrik Ranocha, Mark Wells, Nathan Boyer, Nicholas  
 418 Ritchie, et al. 2024. “Ronisbr/PrettyTables.jl: V2.4.0.” Zenodo. <https://doi.org/10.5281/zenodo.1383553>.
- 419 Christ, Simon, Daniel Schwabeneder, Christopher Rackauckas, Michael Krabbe Borregaard, and Thomas Breloff.  
 420 2023. “Plots.jl – a User Extendable Plotting API for the Julia Programming Language.” <https://doi.org/https://doi.org/10.5334/jors.431>.
- 421 Danisch, Simon, and Julius Krumbiegel. 2021. “Makie.jl: Flexible High-Performance Data Visualization for Julia.”  
 422 *Journal of Open Source Software* 6 (65): 3349. <https://doi.org/10.21105/joss.03349>.
- 423 Du, Yilun, and Igor Mordatch. 2020. “Implicit Generation and Generalization in Energy-Based Models.” <https://arxiv.org/abs/1903.08689>.
- 424 Frankle, Jonathan, and Michael Carbin. 2019. “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural  
 425 Networks.” In *International Conference on Learning Representations*.
- 426 Freiesleben, Timo. 2022. “The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples.”  
 427 *Minds and Machines* 32 (1): 77–109.
- 428 Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- 429 Goodfellow, Ian, Jonathon Shlens, and Christian Szegedy. 2015. “Explaining and Harnessing Adversarial Examples.”  
 430 <https://arxiv.org/abs/1412.6572>.
- 431 Grathwohl, Will, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swer-  
 432 sky. 2020. “Your Classifier Is Secretly an Energy Based Model and You Should Treat It Like One.” In *International  
 433 Conference on Learning Representations*.
- 434 Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. “A Kernel  
 435 Two-Sample Test.” *The Journal of Machine Learning Research* 13 (1): 723–73.
- 436 Guidotti, Riccardo. 2022. “Counterfactual Explanations and How to Find Them: Literature Review and Benchmark-  
 437 ing.” *Data Mining and Knowledge Discovery* 38 (5): 2770–2824. <https://doi.org/10.1007/s10618-022-00831-6>.
- 438 Guo, Hangzhi, Thanh H. Nguyen, and Amulya Yadav. 2023. “CounterNet: End-to-End Training of Prediction Aware  
 439 Counterfactual Explanations.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery*

- 442       and Data Mining, 577--589. KDD '23. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3580305.3599290>.
- 443       Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer New  
444       York. <https://doi.org/10.1007/978-0-387-84858-7>.
- 445       Innes, Michael, Elliot Saba, Keno Fischer, Dhairyा Gandhi, Marco Concetto Rudilosso, Neethu Mariya Joy, Tejan  
446       Karmali, Avik Pal, and Viral Shah. 2018. "Fashionable Modelling with Flux." <https://arxiv.org/abs/1811.01457>.
- 447       Innes, Mike. 2018. "Flux: Elegant Machine Learning with Julia." *Journal of Open Source Software* 3 (25): 602.  
448       <https://doi.org/10.21105/joss.00602>.
- 449       Joshi, Shalmali, Oluwasanmi Koyejo, Warut Vigitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. "Towards Realistic  
450       Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems." <https://arxiv.org/abs/1907.09615>.
- 451       Kaggle. 2011. "Give Me Some Credit, Improve on the State of the Art in Credit Scoring by Pre-  
452       dicting the Probability That Somebody Will Experience Financial Distress in the Next Two Years." <https://www.kaggle.com/c/GiveMeSomeCredit>; Kaggle.
- 453       Kolter, Zico. 2023. "Keynote Addresses: SaTML 2023 ." In *2023 IEEE Conference on Secure and Trustworthy  
454       Machine Learning (SaTML)*. Los Alamitos, CA, USA: IEEE Computer Society. [https://doi.org/10.1109/SaTML5  
456       4575.2023.00009](https://doi.org/10.1109/SaTML5<br/>455       4575.2023.00009).
- 457       Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. "Simple and Scalable Predictive Un-  
458       certainty Estimation Using Deep Ensembles." In *Proceedings of the 31st International Conference on Neural  
459       Information Processing Systems*, 6405–16. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.
- 460       LeCun, Yann. 1998. "The MNIST database of handwritten digits." <http://yann.lecun.com/exdb/mnist/>.
- 461       Lippe, Phillip. 2024. "UvA Deep Learning Tutorials." <https://uvadlc-notebooks.readthedocs.io/en/latest/>.
- 462       Luu, Hoai Linh, and Naoya Inoue. 2023. "Counterfactual Adversarial Training for Improving Robustness of Pre-  
463       trained Language Models." In *Proceedings of the 37th Pacific Asia Conference on Language, Information and  
464       Computation*, 881–88. ACL. <https://aclanthology.org/2023.paclic-1.88>.
- 465       McGregor, Sean. 2021. "Preventing repeated real world AI failures by cataloging incidents: The AI incident database."  
466       In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:15458–63. 17.
- 467       Murphy, Kevin P. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press.
- 468       O'Neil, Cathy. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*.  
469       Crown.
- 470       Pace, R Kelley, and Ronald Barry. 1997. "Sparse Spatial Autoregressions." *Statistics & Probability Letters* 33 (3):  
471       291–97. [https://doi.org/10.1016/s0167-7152\(96\)00140-x](https://doi.org/10.1016/s0167-7152(96)00140-x).
- 472       Pawelczyk, Martin, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. "Exploring  
473       Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis."  
474       In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, edited by Gustau  
475       Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, 151:4574–94. Proceedings of Machine Learning Research.  
476       PMLR. <https://proceedings.mlr.press/v151/pawelczyk22a.html>.
- 477       Poyiadzi, Rafael, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. "FACE: Feasible and  
478       Actionable Counterfactual Explanations." In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*,  
479       344–50.
- 480       Ross, Alexis, Himabindu Lakkaraju, and Osbert Bastani. 2024. "Learning Models for Actionable Recourse." In  
481       *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS '21. Red  
482       Hook, NY, USA: Curran Associates Inc.
- 483       Sauer, Axel, and Andreas Geiger. 2021. "Counterfactual Generative Networks." <https://arxiv.org/abs/2101.06046>.
- 484       Schut, Lisa, Oscar Key, Rory McGrath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. 2021. "Generating  
485       Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties." In  
486       *International Conference on Artificial Intelligence and Statistics*, 1756–64. PMLR.
- 487       Sharma, Shubham, Jette Henderson, and Joydeep Ghosh. 2020. "CERTIFAI: A Common Framework to Provide  
488       Explanations and Analyse the Fairness and Robustness of Black-box Models." In *Proceedings of the AAAI/ACM  
489       Conference on AI, Ethics, and Society*, 166–72. AIES '20. New York, NY, USA: Association for Computing  
490       Machinery. <https://doi.org/10.1145/3375627.3375812>.
- 491       Spooner, Thomas, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, and Daniele Magazzeni. 2021. "Counter-  
492       factual Explanations for Arbitrary Regression Models." <https://arxiv.org/abs/2106.15212>.
- 493       Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.  
494       2014. "Intriguing Properties of Neural Networks." <https://arxiv.org/abs/1312.6199>.
- 495       Teh, Yee Whye, Max Welling, Simon Osindero, and Geoffrey E. Hinton. 2003. "Energy-Based Models for Sparse  
496       Overcomplete Representations." *J. Mach. Learn. Res.* 4 (null): 1235–60.

- 499 Tenev, Damien, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. “Learning What Makes a Difference from  
 500 Counterfactual Examples and Gradient Supervision.” In *Computer Vision - ECCV 2020*, 580–99. Berlin, Heidelberg:  
 501 Springer-Verlag. [https://doi.org/10.1007/978-3-030-58607-2\\_34](https://doi.org/10.1007/978-3-030-58607-2_34).
- 502 Venkatasubramanian, Suresh, and Mark Alfano. 2020. “The Philosophical Basis of Algorithmic Recourse.” In *Pro-  
 503 ceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 284–93. FAT\* ’20. New York,  
 504 NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3351095.3372876>.
- 505 Wachter, Sandra, Brent Mittelstadt, and Chris Russell. 2017. “Counterfactual Explanations Without Opening the Black  
 506 Box: Automated Decisions and the GDPR.” *Harv. JL & Tech.* 31: 841. <https://doi.org/10.2139/ssrn.3063289>.
- 507 Wilson, Andrew Gordon. 2020. “The Case for Bayesian Deep Learning.” <https://arxiv.org/abs/2001.10995>.
- 508 Wu, Tongshuang, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. “Polyjuice: Generating Counterfactuals  
 509 for Explaining, Evaluating, and Improving Models.” In *Proceedings of the 59th Annual Meeting of the Associa-  
 510 tion for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing  
 511 (Volume 1: Long Papers)*, 6707–23. ACL. <https://doi.org/10.18653/v1/2021.acl-long.523>.
- 512 Yeh, I-Cheng. 2016. “Default of Credit Card Clients.” UCI Machine Learning Repository.
- 513 Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. “Understanding Deep  
 514 Learning (Still) Requires Rethinking Generalization.” *Commun. ACM* 64 (3): 107–15. <https://doi.org/10.1145/3446776>.

516 **Appendix A Notation**

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

- 518 •  $y^+$ : The target class and also the index of the target class.
- 519 •  $y^-$ : The non-target class and also the index of non-the target class.
- 520 •  $\mathbf{x}$ : a single training sample.
- 521 •  $\mathbf{x}'$ : a counterfactual.
- 522 •  $\mathbf{x}^+$ : a training sample in the target class (ground-truth).
- 523 •  $\mathbf{y}^+$ : The one-hot encoded output vector for the target class.
- 524 •  $\theta$ : Model parameters (unspecified).
- 525 •  $\Theta$ : Matrix of parameters.
- 526 •  $\mathbf{M}(\cdot)$ : linear predictions (logits) of the classifier.

527 **A.1 Other Technical Details**

528 Maximum mean discrepancy is defined as follows,

$$\begin{aligned} \text{MMD}(X', \tilde{X}') &= \frac{1}{m(m-1)} \sum_{i=1}^m \sum_{j \neq i}^m k(x_i, x_j) \\ &\quad + \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i}^n k(\tilde{x}_i, \tilde{x}_j) \\ &\quad - \frac{2}{mn} \sum_{i=1}^m \sum_{j=1}^n k(x_i, \tilde{x}_j) \end{aligned} \tag{6}$$

529 where  $k(\cdot, \cdot)$  is a kernel function (Gretton et al. 2012). We make use of a Gaussian kernel with a constant length-scale  
530 parameter of 0.5. In our implementation, Equation 6 is by default applied to the entire subset of the training data for  
531 which  $y = y^+$ .

532 **Appendix B Technical Details of Our Approach**

533 **B.1 Generating Counterfactuals through Gradient Descent**

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

536 **B.1.1 Background**

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

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

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

543 For all generators used in this work we use standard logit crossentropy loss for  $\text{ylloss}(\cdot)$ . All generators also penalize  
544 the distance ( $\ell_1$ -norm) of counterfactuals from their original factual state. For *Generic* and *ECCo*, we have  $\mathcal{Z} := \mathcal{X}$   
545 and  $g(\mathbf{z}) = g(\mathbf{z})^{-1} = \mathbf{z}$ , that is counterfactual are searched directly in the feature space. Conversely, *REVISE* traverses  
546 the latent space of a variational autoencoder (VAE) fitted to the training data, where  $g(\cdot)$  corresponds to the decoder  
547 (Joshi et al. 2019). In addition to the distance penalty, *ECCo* uses an additional penalty component that regularizes  
548 the energy associated with the counterfactual,  $\mathbf{x}'$  (Altmeyer et al. 2024).

549 **B.1.2 Convergence**

550 An important consideration when generating counterfactual explanations using gradient-based methods is how to  
551 define convergence. Two common choices are to 1) perform gradient descent over a fixed number of iterations  $T$ , or  
552 2) conclude the search as soon as the predicted probability for the target class has reached a pre-determined threshold,

553  $\tau: \mathcal{S}(\mathbf{M}_\theta(\mathbf{x}'))[y^+] \geq \tau$ . We prefer the latter for our purposes, because it explicitly defines convergence in terms of the  
 554 black-box model,  $\mathbf{M}(\mathbf{x})$ .

555 Defining convergence in this way allows for a more intuitive interpretation of the resulting counterfactual outcomes  
 556 than with fixed  $\bar{T}$ . Specifically, it allows us to think of counterfactuals as explaining ‘high-confidence’ predictions by  
 557 the model for the target class  $y^+$ . Depending on the context and application, different choices of  $\tau$  can be considered  
 558 as representing ‘high-confidence’ predictions.

## 559 B.2 Protecting Mutability Constraints with Linear Classifiers

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

564 *Proof.* Let  $d_{\text{mtbl}}$  and  $d_{\text{immmtbl}}$  denote some mutable and immutable feature, respectively. Suppose that  $\mu_{y^-, d_{\text{immmtbl}}} <$   
 565  $\mu_{y^+, d_{\text{immmtbl}}}$  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,  
 566 we assume that the immutable feature tends to take lower values for samples in the non-target class  $y^-$  than in the  
 567 target class  $y^+$ . We assume the opposite to hold for the mutable feature.

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

$$\log \frac{p(k|\mathbf{x})}{p(m|\mathbf{x})} = \mathbf{x}^\top \Sigma^{-1} (\mu_k - \mu_m) + \text{const} \quad (7)$$

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

$$\log \frac{p(k|\mathbf{x})}{p(m|\mathbf{x})} = \sum_{d=1}^D \frac{\mu_{k,d} - \mu_{m,d}}{\sigma_d^2} x_d + \text{const} \quad (8)$$

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

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

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

573 Based on Equation 8 and Equation 9 we can identify that  $(\mu_{k,d} - \mu_{m,d}) \propto (\theta_{k,d} - \theta_{m,d})$  under the assumptions we  
 574 made above. Hence, we have that  $(\theta_{y^-, d_{\text{immmtbl}}} - \theta_{y^+, d_{\text{immmtbl}}}) < 0$  and  $(\theta_{y^-, d_{\text{mtbl}}} - \theta_{y^+, d_{\text{mtbl}}}) > 0$

575 Let  $\mathbf{x}'$  denote some randomly chosen individual from class  $y^-$  and let  $y^+ \sim p(y)$  denote the randomly chosen target  
 576 class. Then the partial derivative of the contrastive divergence penalty Equation 2 with respect to coefficient  $\theta_{y^+, d}$  is  
 577 equal to

$$\frac{\partial}{\partial \theta_{y^+, d}} (\text{div}(\mathbf{x}^+, \mathbf{x}', \mathbf{y}; \theta)) = \frac{\partial}{\partial \theta_{y^+, d}} ((-\mathbf{M}_\theta(\mathbf{x}^+)[y^+]) - (-\mathbf{M}_\theta(\mathbf{x}') [y^+])) = x'_d - x_d^+ \quad (10)$$

578 and equal to zero everywhere else.

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

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

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

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

### 591 B.3 Domain Constraints

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

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

$$\begin{aligned} x_d^{\text{LB}} &= \arg \min_{x_d} \{\mu_d - n_{\sigma_d} \sigma_d, \arg \min_{x_d} x_d\} \\ x_d^{\text{UB}} &= \arg \max_{x_d} \{\mu_d + n_{\sigma_d} \sigma_d, \arg \max_{x_d} x_d\} \end{aligned} \quad (11)$$

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

### 602 B.4 Training Hyperparameters

603 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
    - \*  $\lambda_{\text{cst}}$ : 0.001
    - \*  $\lambda_{\text{egy}}$ : 5.0
    - \* Learning Rate: 0.25
    - \* Maximum Iterations: 30
    - \* Optimizer: `sgd`
    - \* Type: ECCo
  - $\lambda_{\text{adv}}$ : 0.25
  - $\lambda_{\text{clf}}$ : 1.0
  - $\lambda_{\text{div}}$ : 0.5
  - $\lambda_{\text{reg}}$ : 0.1
  - Learning Rate: 0.001
  - No. Counterfactuals: 1000
  - No. Epochs: 100

- Objective: full
- Optimizer: adam

605

606 **B.5 Evaluation Details**

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

613 **B.5.1 Robust Accuracy**

614 To evaluate robust accuracy (Acc.\*), we use the Fast Gradient Sign Method (FGSM) to perturb test samples (Goodfellow,  
 615 Shlens, and Szegedy 2015). For the main results, we have set the perturbation size to  $\epsilon = 0.03$ . We have also  
 616 tested other perturbation sizes, as well as randomly perturbed data. Although not reported here, we have consistently  
 617 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
    - \*  $\lambda_{\text{cst}}$ : 0.001
    - \*  $\lambda_{\text{egy}}$ : 5.0
    - \* Learning Rate: 0.25
    - \* Maximum Iterations: 30
    - \* Optimizer: sgd
    - \* Type: ECCo
  - Maximum Iterations: 50
  - No. Individuals: 100
  - No. Runs: 5

618

619 **Appendix C Details on Main Experiments**620 **C.1 Final Hyperparameters**

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

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	$\lambda_{\text{reg}}$
Adult	26049	5010	1000	none	0.75	5000	0.25
CH	16504	3101	1000	none	0.5	5000	0.25
Circ	3600	600	30	none	0.5	1000	0.5
Cred	10617	1923	1000	none	0.5	5000	0.25
GMSC	13371	2474	1000	none	0.5	5000	0.5
LS	3600	600	30	none	0.5	1000	0.01
MNIST	11000	2000	1000	(-1.0, 1.0)	0.5	5000	0.01
Moon	3600	600	30	none	0.9	1000	0.25
OL	3600	600	30	none	0.5	1000	0.25

624 **C.2 Qualitative Findings for Image Data**

625 Figure 2 shows much more plausible (faithful) counterfactuals for a model with CT than the model with conventional  
626 training (Figure 3).

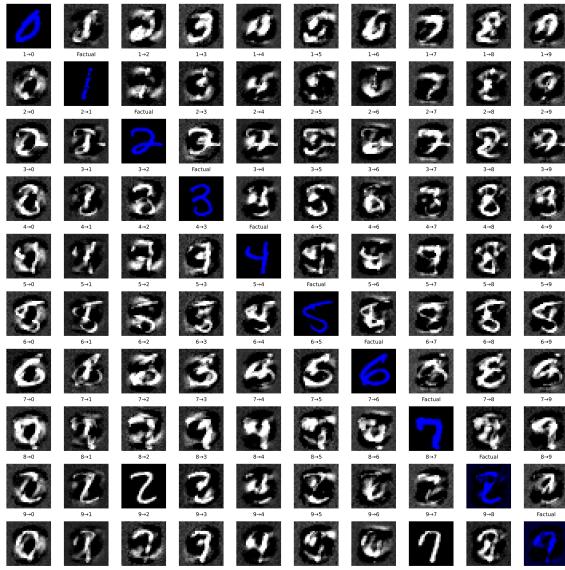


Figure 2: Counterfactual images for *MLP* with counterfactual training. Factual images are shown on the diagonal, with the corresponding counterfactual for each target class (columns) in that same row. The underlying generator, *ECCo*, aims to generate counterfactuals that are faithful to the model (Altmeier et al. 2024).

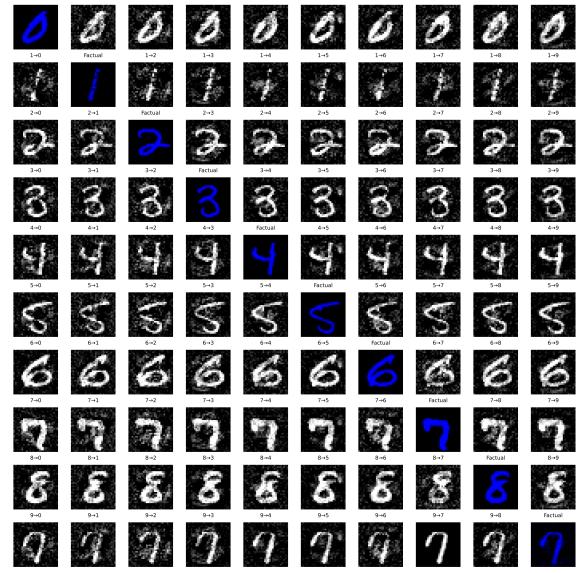


Figure 3: The same setup, factuals, model architecture and generator as in Figure 2, but the model was trained with CT.

627 **Appendix D Grid Searches**

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

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

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

639 **D.1 Evaluation Details**

640 To measure predictive performance, we compute the accuracy and F1-score for all models on test data (Table 3,  
641 Table 4, Table 5). With respect to explanatory performance, we report here our findings for the (im)plausibility and  
642 cost of counterfactuals at test time. Since the computation of our proposed divergence metric (Equation 5) is memory-  
643 intensive, we rely on the distance-based metric for the grid searches. For the counterfactual evaluation, we draw factual  
644 samples from the training data for the grid searches to avoid data leakage with respect to our final results reported in  
645 the body of the paper. Specifically, we want to avoid choosing our default hyperparameters based on results on the  
646 test data. Since we are optimizing for explainability, not predictive performance, we still present test accuracy and  
647 F1-scores.

648 **D.1.1 Predictive Performance**

649 We find that CT is associated with little to no decrease in average predictive performance for our synthetic datasets: test  
 650 accuracy and F1-scores decrease by at most ~1 percentage point, but generally much less (Table 3, Table 4, Table 5).  
 651 Variation across hyperparameters is negligible as indicated by small standard deviations for these metrics across the  
 652 board.

653 **D.1.2 Counterfactual Outcomes**

654 Overall, we find that counterfactual training (CT) achieves its key objectives consistently across all hyperparameter  
 655 settings and also broadly across datasets: plausibility is improved by up to ~60 percent (%) for the *Circles* data (e.g.  
 656 Figure 4), ~25-30% for the *Moons* data (e.g. Figure 6) and ~10-20% for the *Linearly Separable* data (e.g. Figure 5). At  
 657 the same time, the average costs of faithful counterfactuals are reduced in many cases by around ~20-25% for *Circles*  
 658 (e.g. Figure 8) and up to ~50% for *Moons* (e.g. Figure 10). For the *Linearly Separable* data, costs are generally  
 659 increased although typically by less than 10% (e.g. Figure 9), which reflects a common tradeoff between costs and  
 660 plausibility (Altmeyer et al. 2024).

661 We do observe strong sensitivity to certain hyperparameters, with clear manageable patterns. Concerning generator  
 662 parameters, we firstly find that using *REVISE* to generate counterfactuals during training typically yields the worst  
 663 outcomes out of all generators, often leading to a substantial decrease in plausibility. This finding can be attributed to  
 664 the fact that *REVISE* effectively assigns the task of learning plausible explanations from the model itself to a surrogate  
 665 VAE. In other words, counterfactuals generated by *REVISE* are less faithful to the model than *ECCo* and *Generic*, and  
 666 hence we would expect them to be a less effective and, in fact, potentially detrimental role in our training regime.  
 667 Secondly, we observe that allowing for a higher number of maximum steps  $T$  for the counterfactual search generally  
 668 yields better outcomes. This is intuitive, because it allows more counterfactuals to reach maturity in any given iteration.  
 669 Looking in particular at the results for *Linearly Separable*, it seems that higher values for  $T$  in combination with higher  
 670 decision thresholds ( $\tau$ ) yields the best results when using *ECCo*. But depending on the degree of class separability  
 671 of the underlying data, a high decision-threshold can also affect results adversely, as evident from the results for the  
 672 *Overlapping* data (Figure 7): here we find that CT generally fails to achieve its objective because only a tiny proportion  
 673 of counterfactuals ever reaches maturity.

674 Regarding penalty strengths, we find that the strength of the energy regularization,  $\lambda_{\text{reg}}$  is a key hyperparameter, while  
 675 sensitivity with respect to  $\lambda_{\text{div}}$  and  $\lambda_{\text{adv}}$  is much less evident. In particular, we observe that not regularizing energy  
 676 enough or at all typically leads to poor performance in terms of decreased plausibility and increased costs, in particular  
 677 for *Circles* (Figure 12), *Linearly Separable* (Figure 13) and *Overlapping* (Figure 15). High values of  $\lambda_{\text{reg}}$  can increase  
 678 the variability in outcomes, in particular when combined with high values for  $\lambda_{\text{div}}$  and  $\lambda_{\text{adv}}$ , but this effect is less  
 679 pronounced.

680 Finally, concerning other hyperparameters we observe that the effectiveness and stability of CT is positively associated  
 681 with the number of counterfactuals generated during each training epoch, in particular for *Circles* (Figure 20) and  
 682 *Moons* (Figure 22). We further find that a higher number of training epochs is beneficial as expected, where we tested  
 683 training models for 50 and 100 epochs. Interestingly, we find that it is not necessary to employ CT during the entire  
 684 training phase to achieve the desired improvements in explainability: specifically, we have tested training models  
 685 conventionally during the first half of training before switching to CT after this initial burn-in period.

686 **D.2 Generator Parameters**

687 The hyperparameter grid with varying generator parameters during training is shown in Note 3. The corresponding  
 688 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
  - $\lambda_{\text{egy}}$ : 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`

689

Note 4: Evaluation Phase

- Generator Parameters:
  - $\lambda_{\text{egy}}$ : 0.1, 0.5, 1.0, 5.0, 10.0

690

691 **D.2.1 Predictive Performance**

692 Predictive performance measures for this grid search are shown in Table 3.

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

Dataset	Variable	Objective	Mean	Std
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
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

693 **D.2.2 Plausibility**

694 The results with respect to the plausibility measure are shown in Figure 4 to Figure 7.

695 **D.2.3 Cost**

696 The results with respect to the cost measure are shown in Figure 8 to Figure 11.

697 **D.3 Penalty Strengths**

698 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:
  - $\lambda_{\text{adv}}$ : 0.1, 0.25, 1.0
  - $\lambda_{\text{div}}$ : 0.01, 0.1, 1.0
  - $\lambda_{\text{reg}}$ : 0.0, 0.01, 0.1, 0.25, 0.5
  - Objective: `full`, `vanilla`

700

Note 6: Evaluation Phase

- Generator Parameters:
  - $\lambda_{\text{egy}}$ : 0.1, 0.5, 1.0, 5.0, 10.0

701

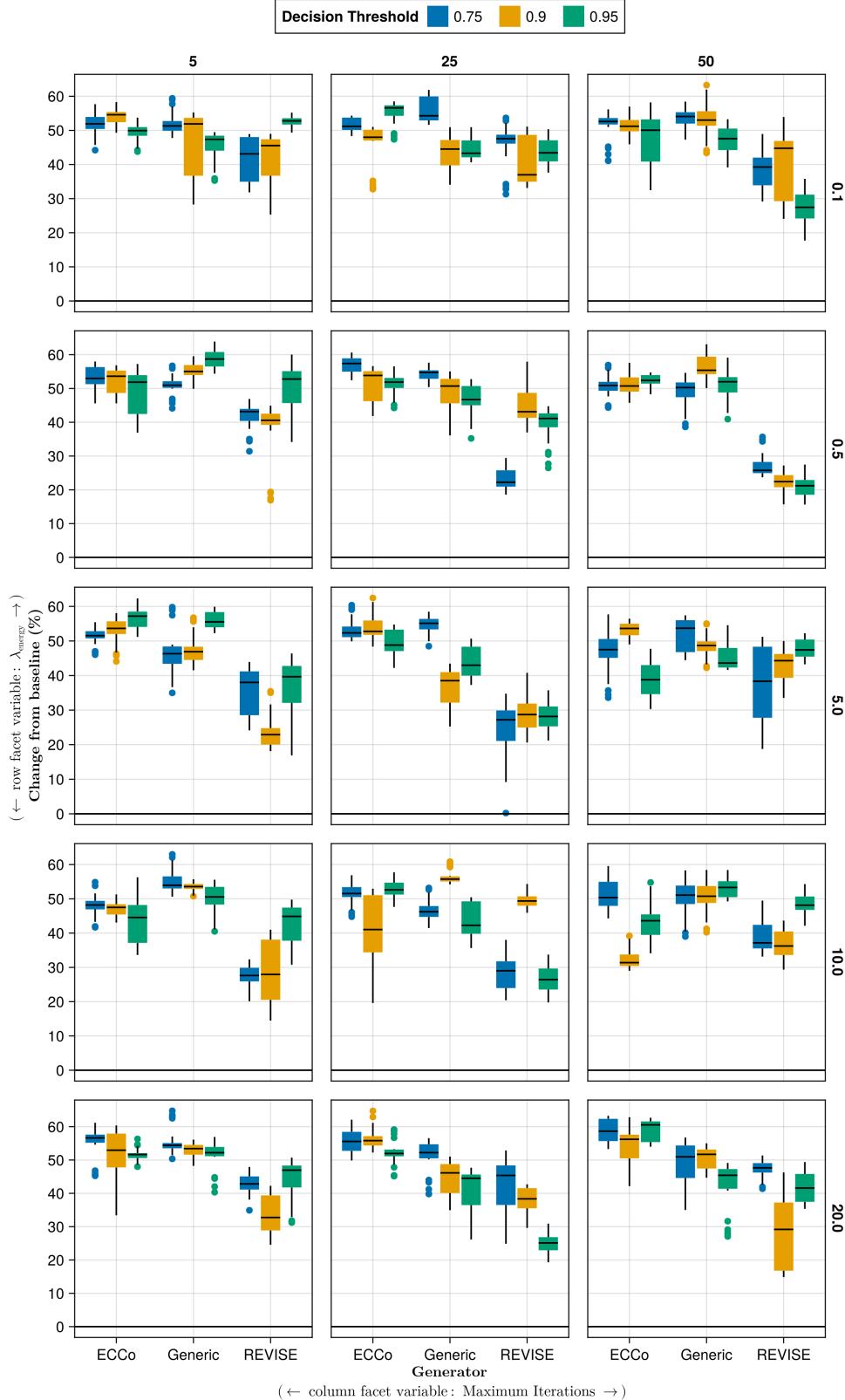


Figure 4: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

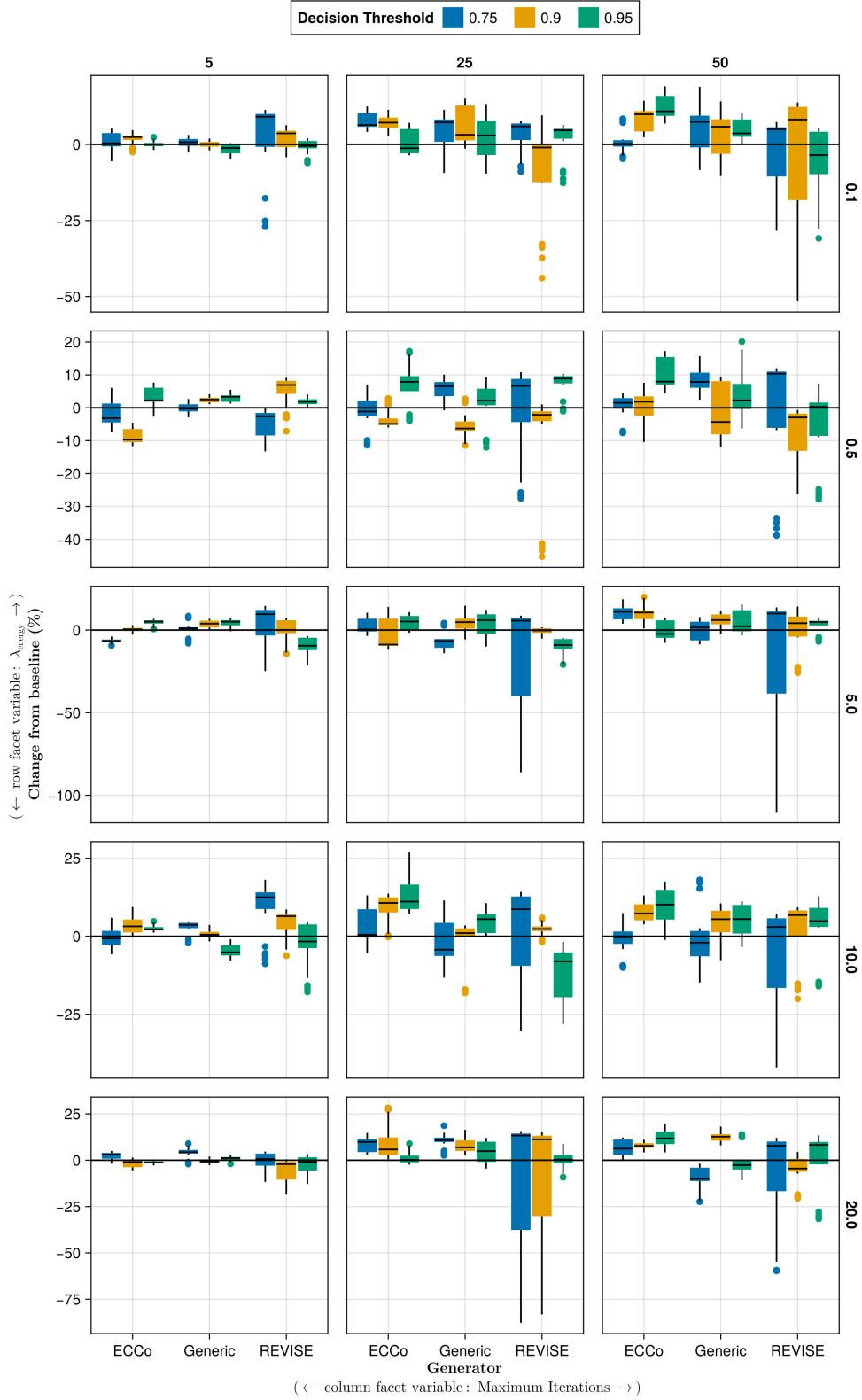


Figure 5: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

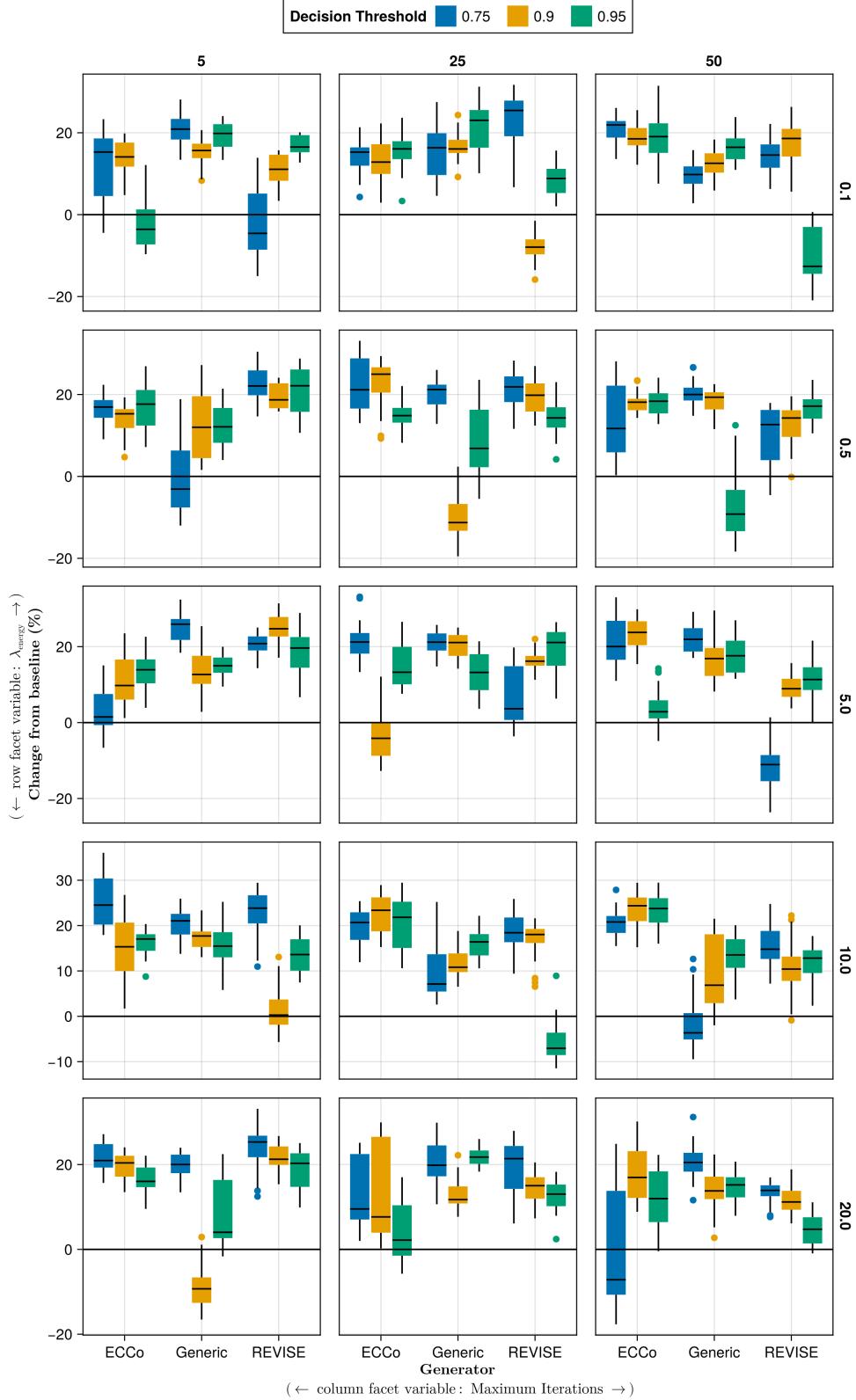


Figure 6: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

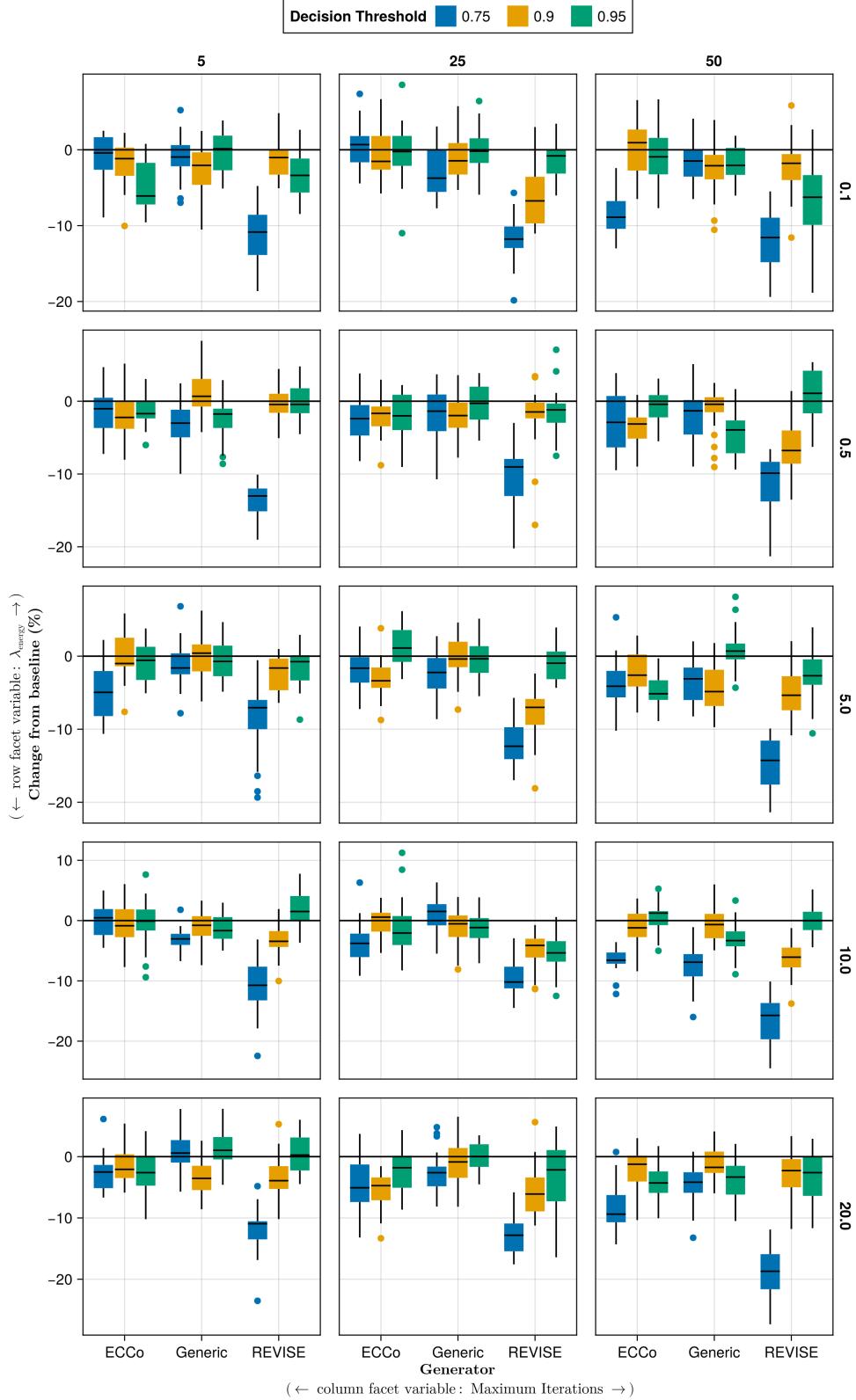


Figure 7: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

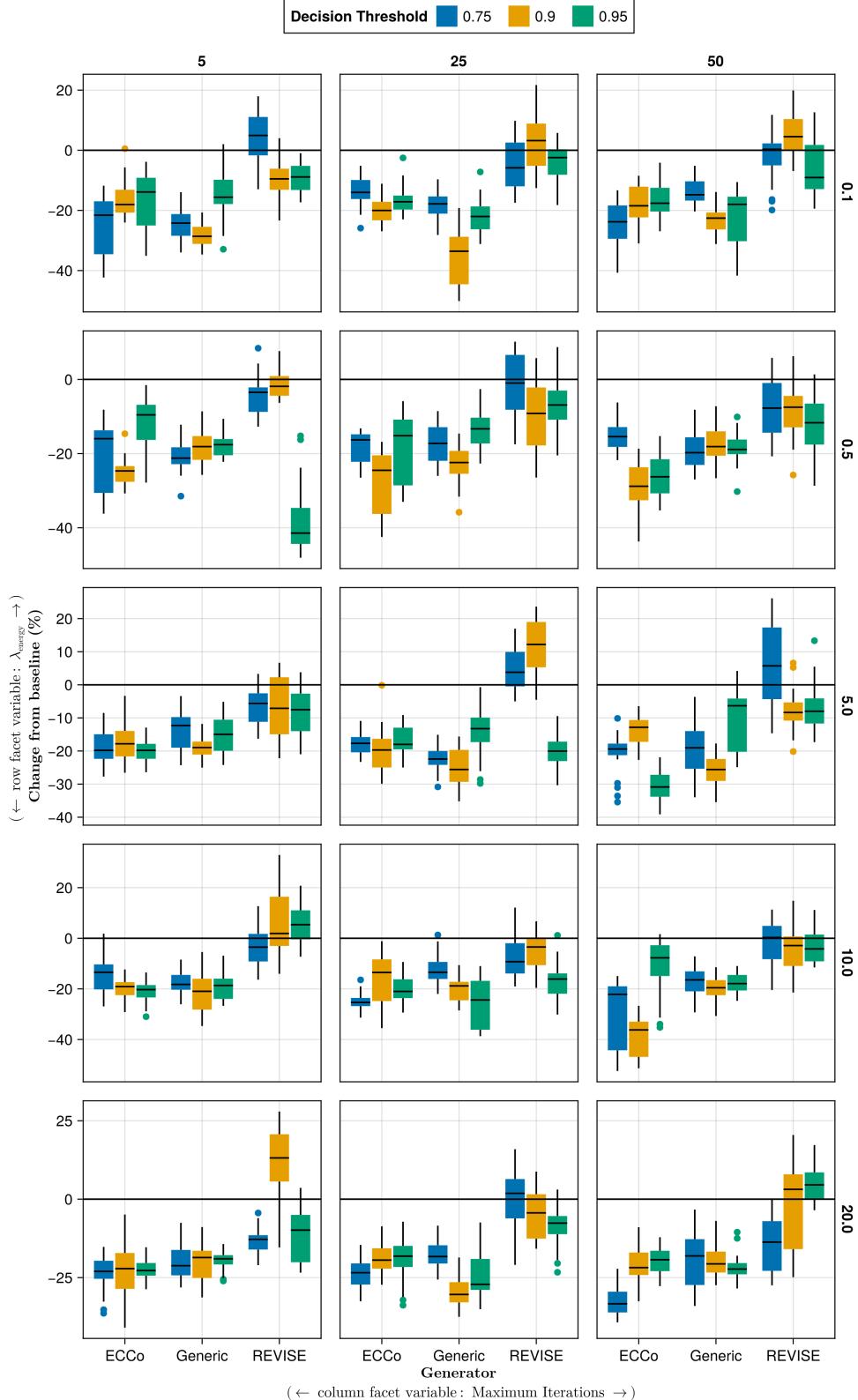


Figure 8: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

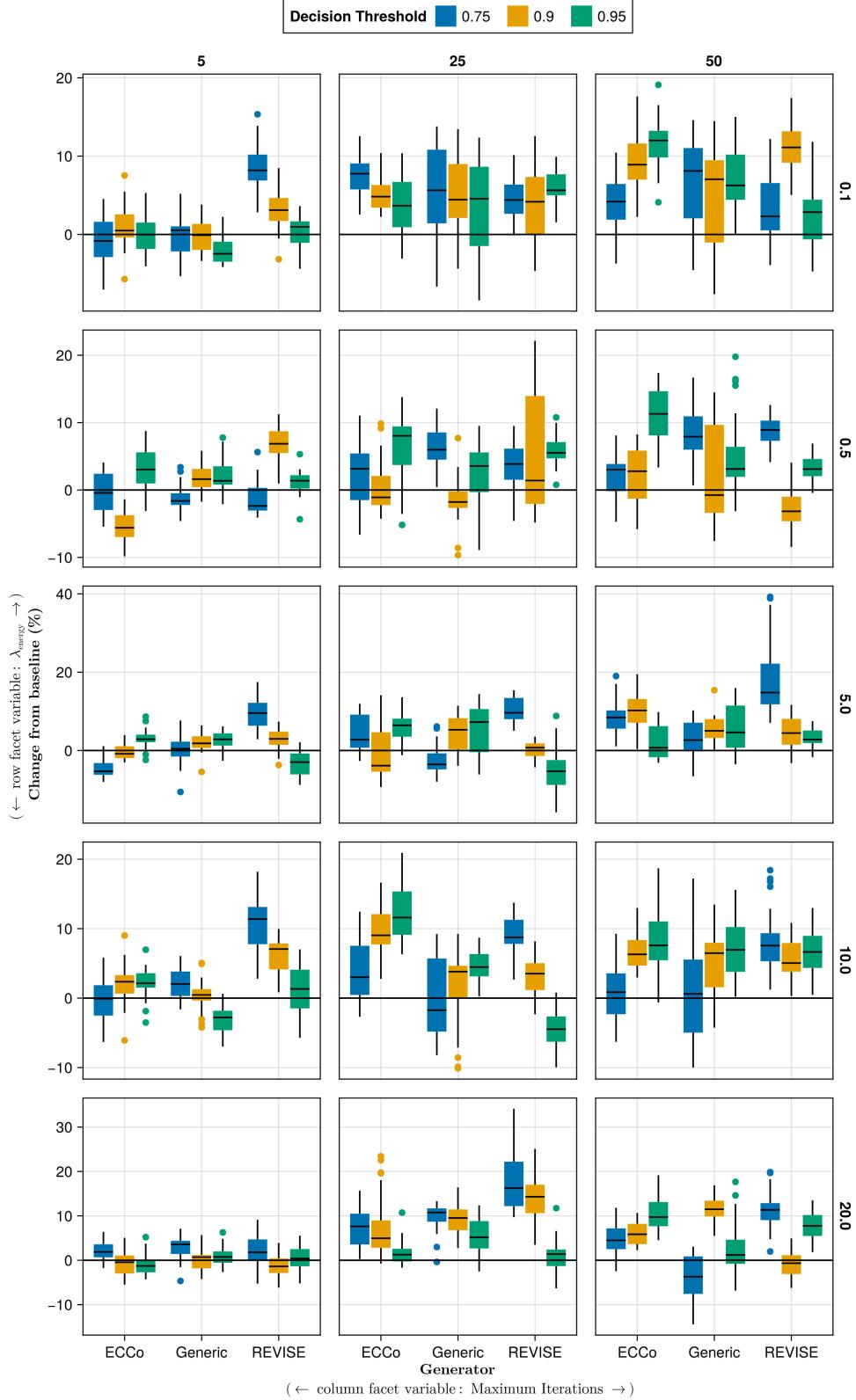


Figure 9: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

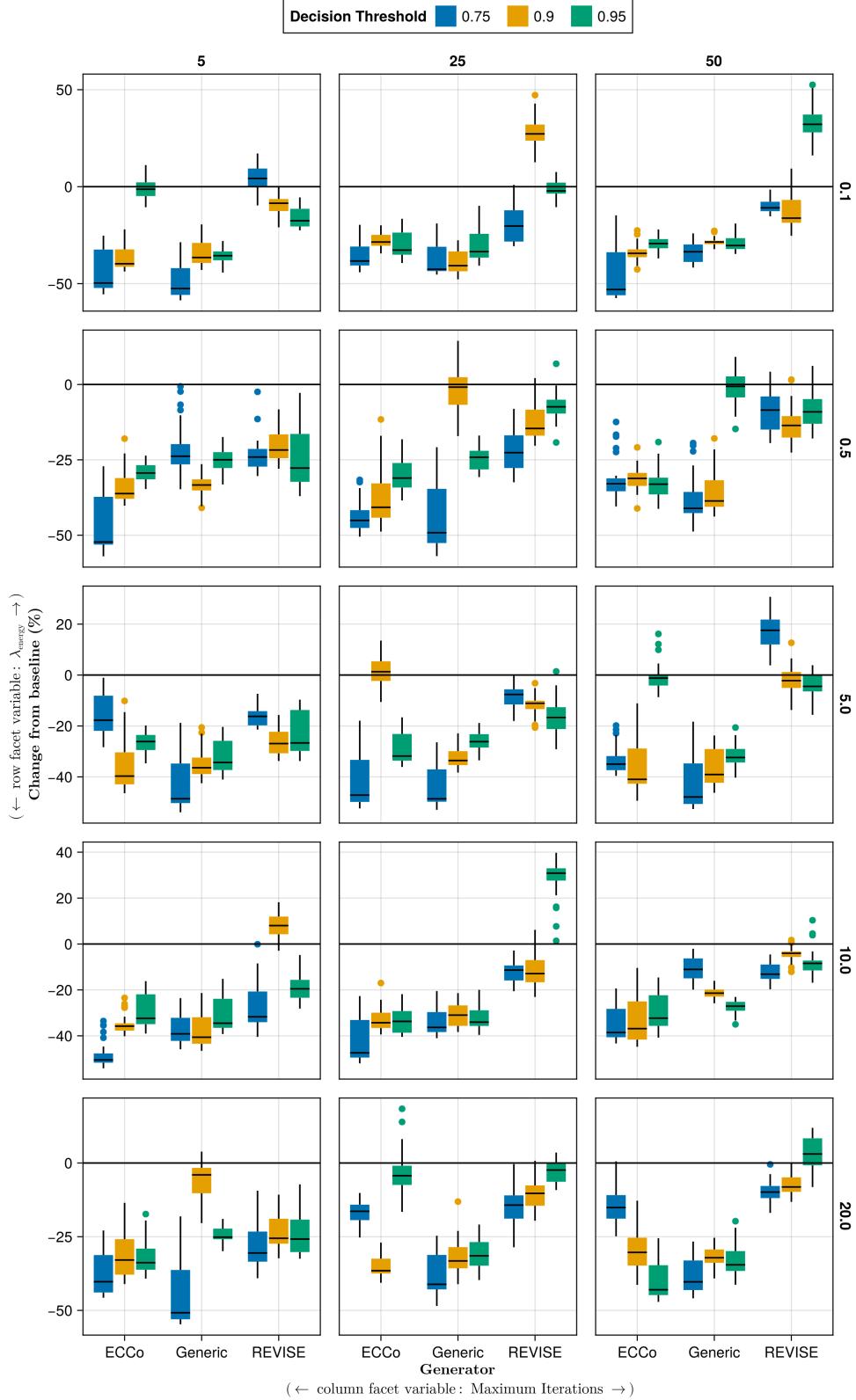


Figure 10: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

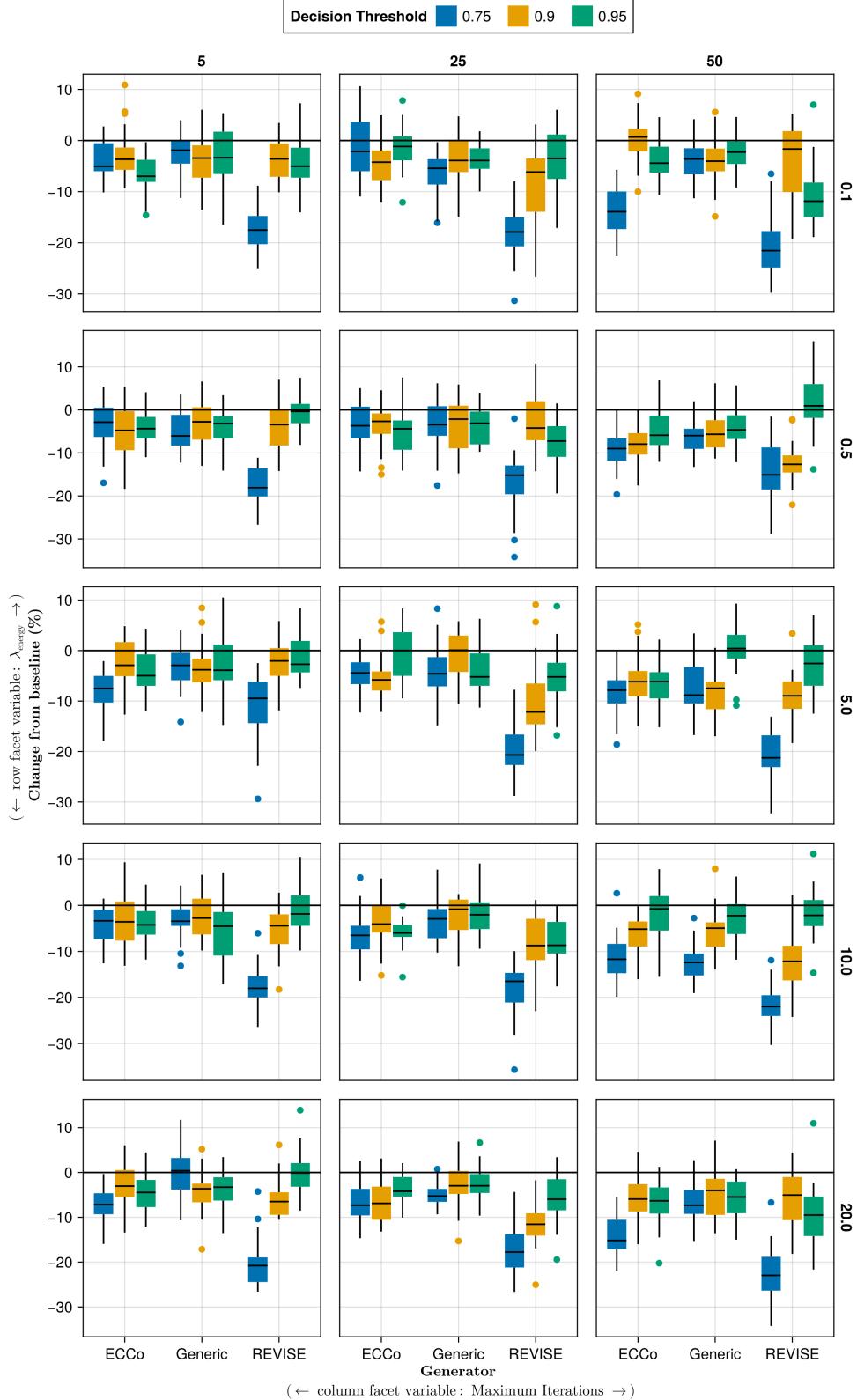


Figure 11: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric (Wachter, Mittelstadt, and Russell 2017). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

702 **D.3.1 Predictive Performance**

703 Predictive performance measures for this grid search are shown in Table 4.

Table 4: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 5) and evaluation-phase parameters (Note 6).

Dataset	Variable	Objective	Mean	Std
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
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

704 **D.3.2 Plausibility**

705 The results with respect to the plausibility measure are shown in Figure 12 to Figure 15.

706 **D.3.3 Cost**

707 The results with respect to the cost measure are shown in Figure 16 to Figure 19.

708 **D.4 Other Parameters**709 The hyperparameter grid with other varying training parameters is shown in Note 7. The corresponding evaluation  
710 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`

711

## Note 8: Evaluation Phase

- Generator Parameters:
  - $\lambda_{\text{egy}}$ : 0.1, 0.5, 1.0, 5.0, 10.0

712

713 **D.4.1 Predictive Performance**

714 Predictive performance measures for this grid search are shown in Table 5.

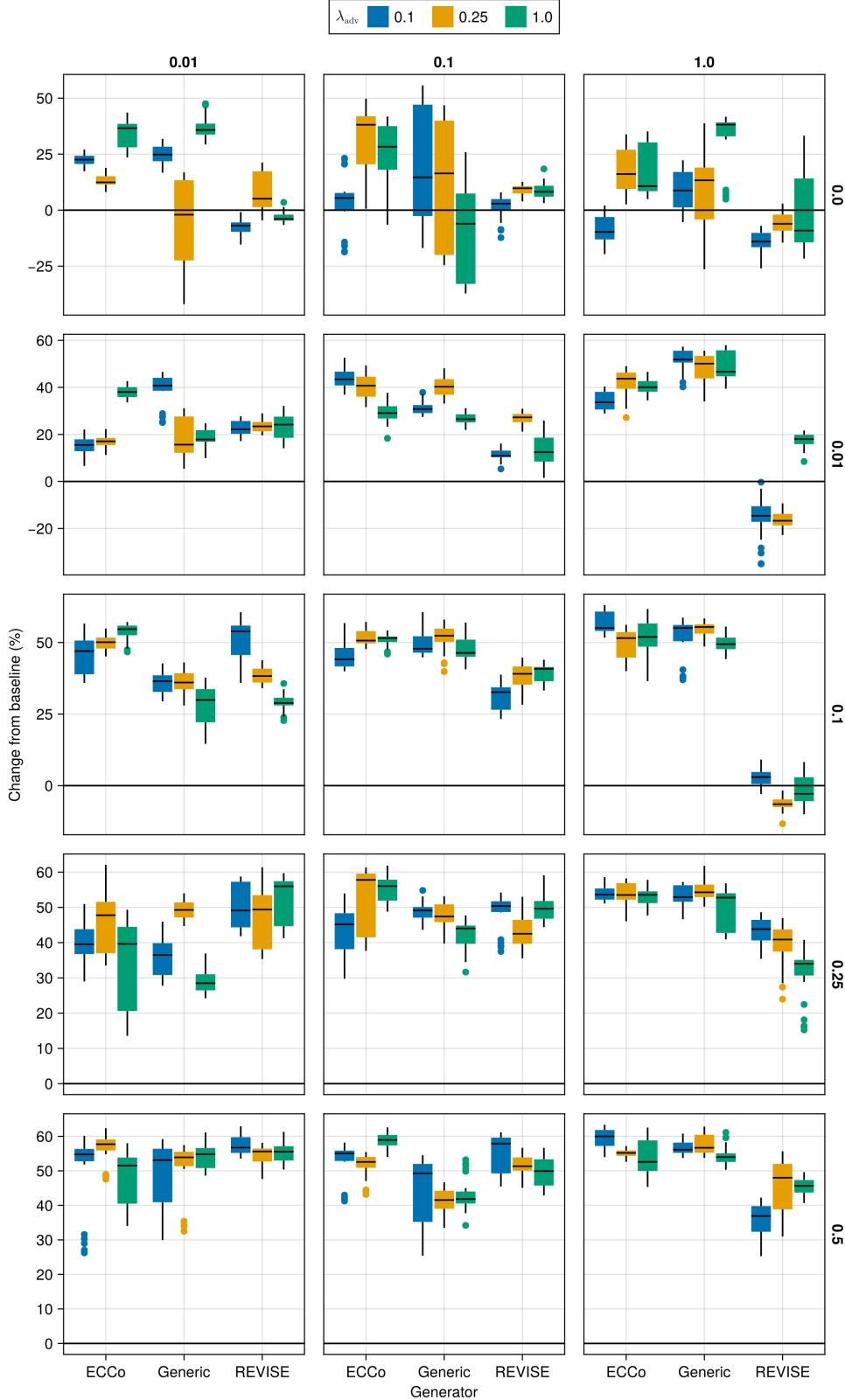


Figure 12: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

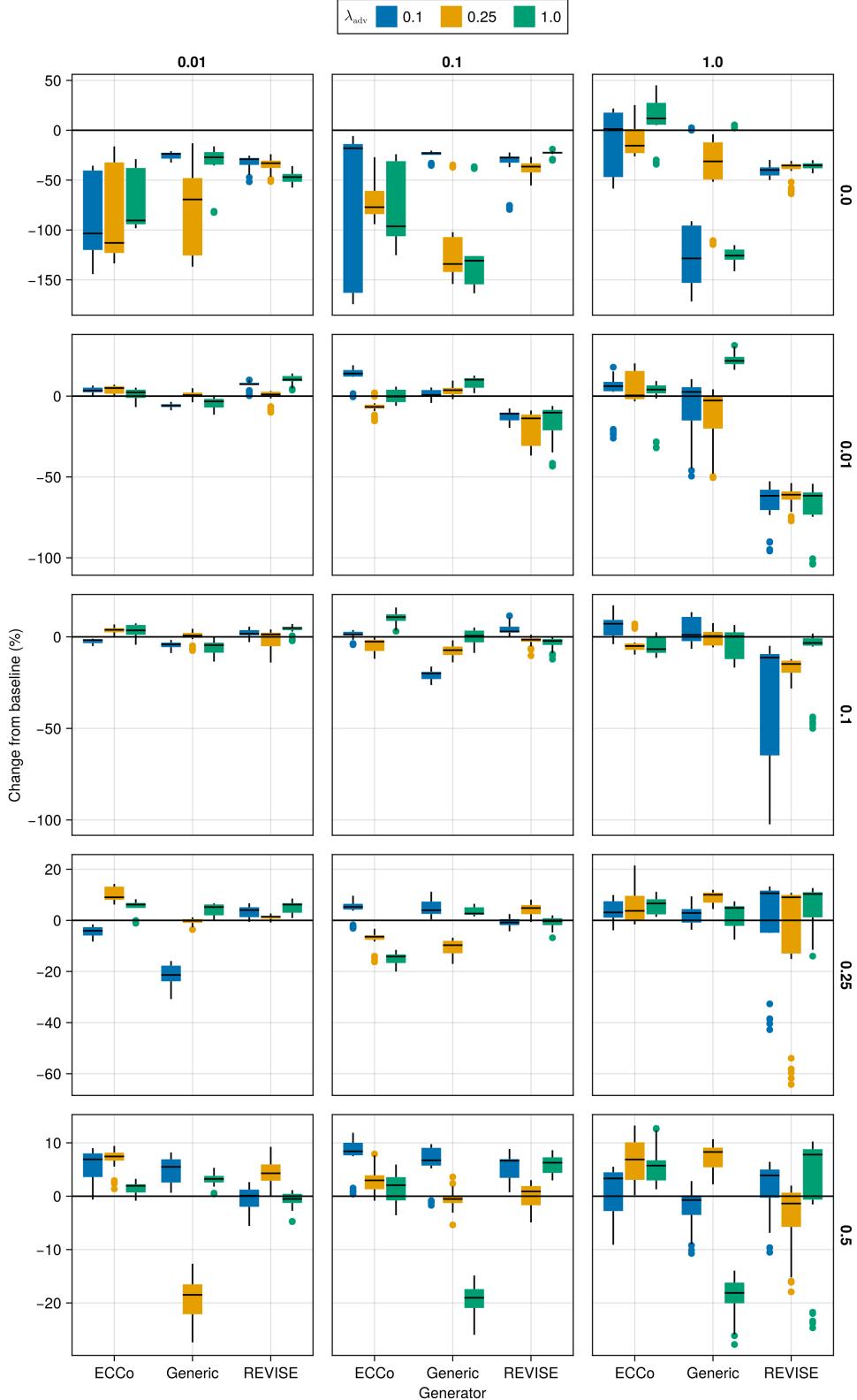


Figure 13: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

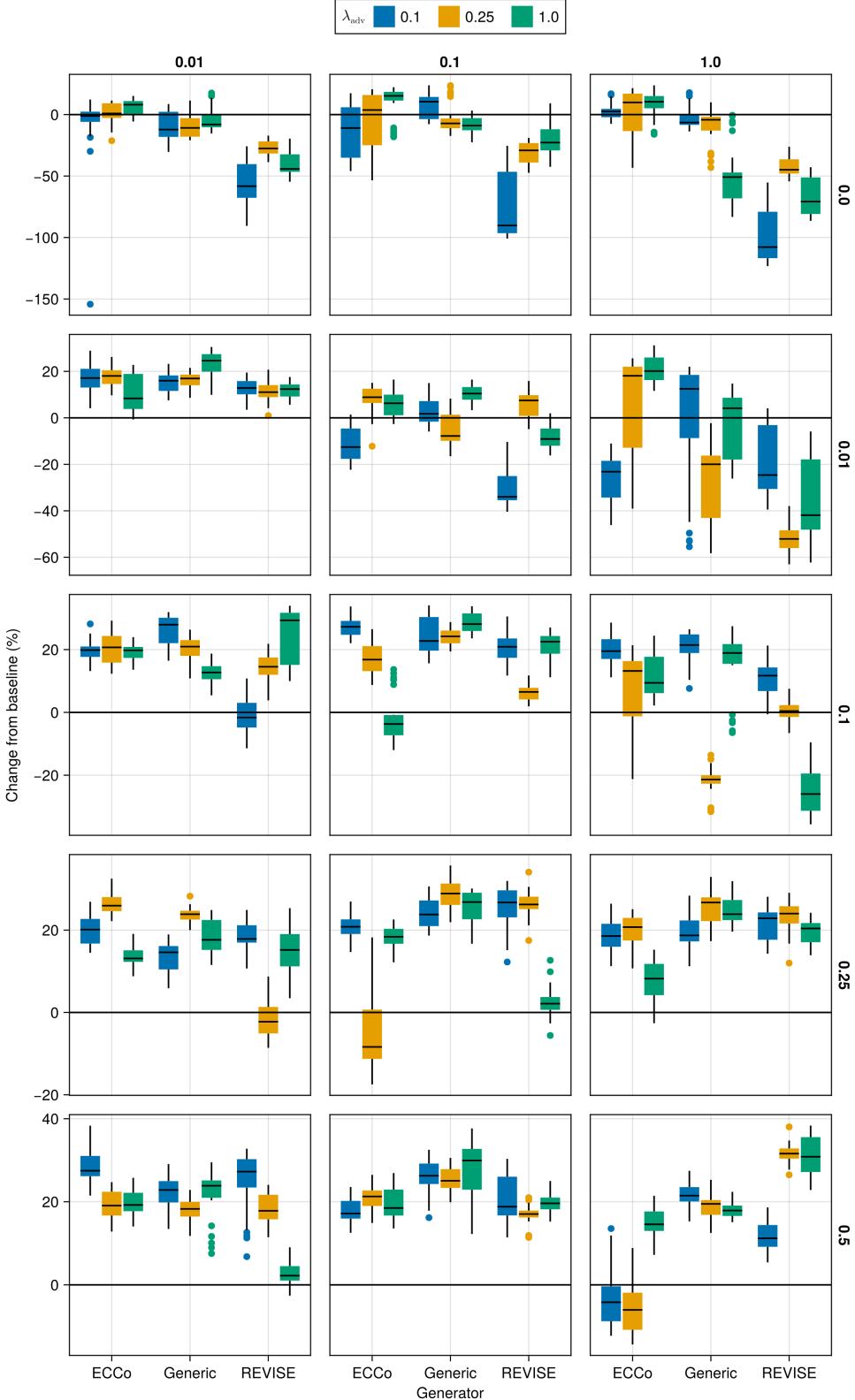


Figure 14: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

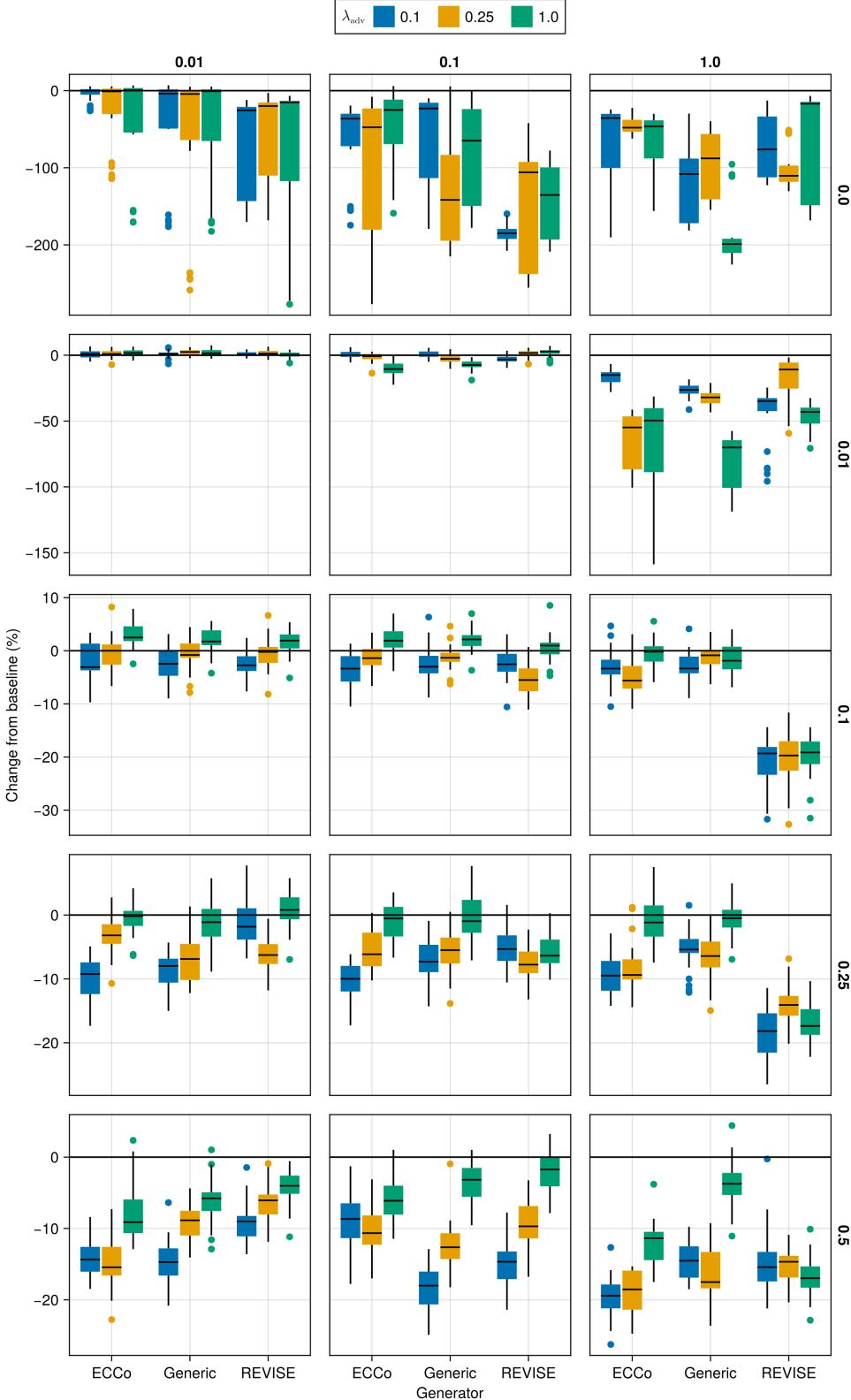


Figure 15: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

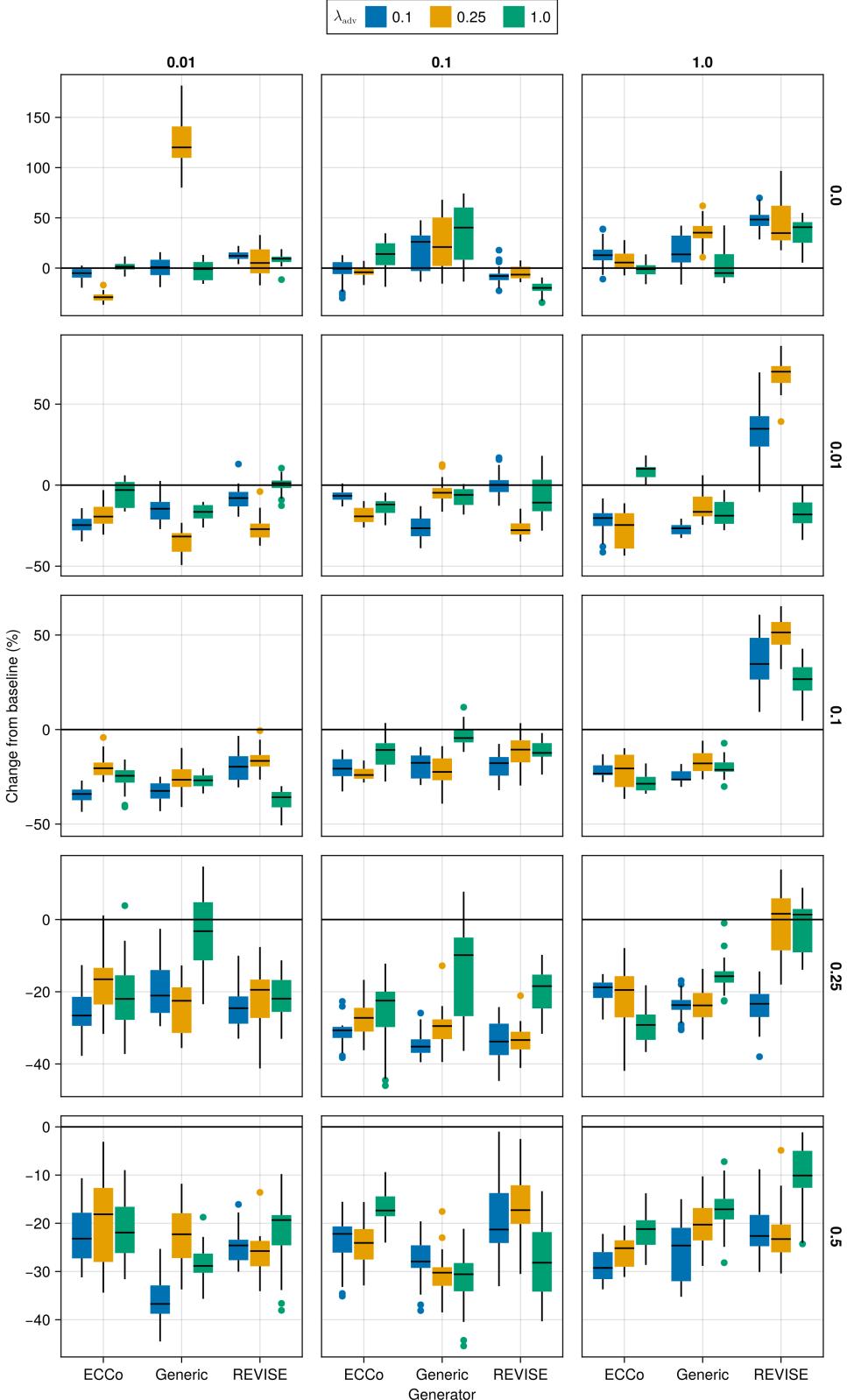


Figure 16: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

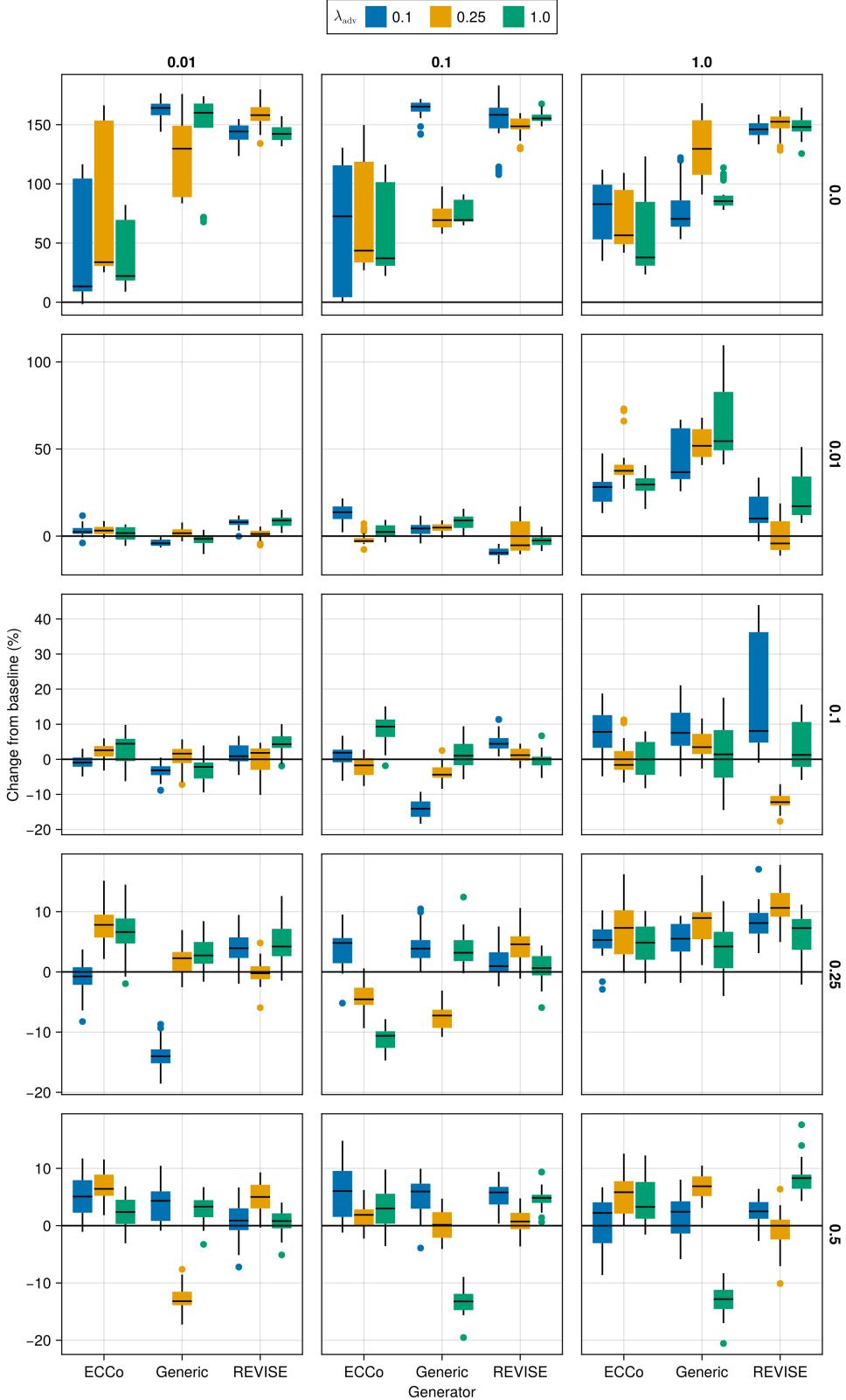


Figure 17: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

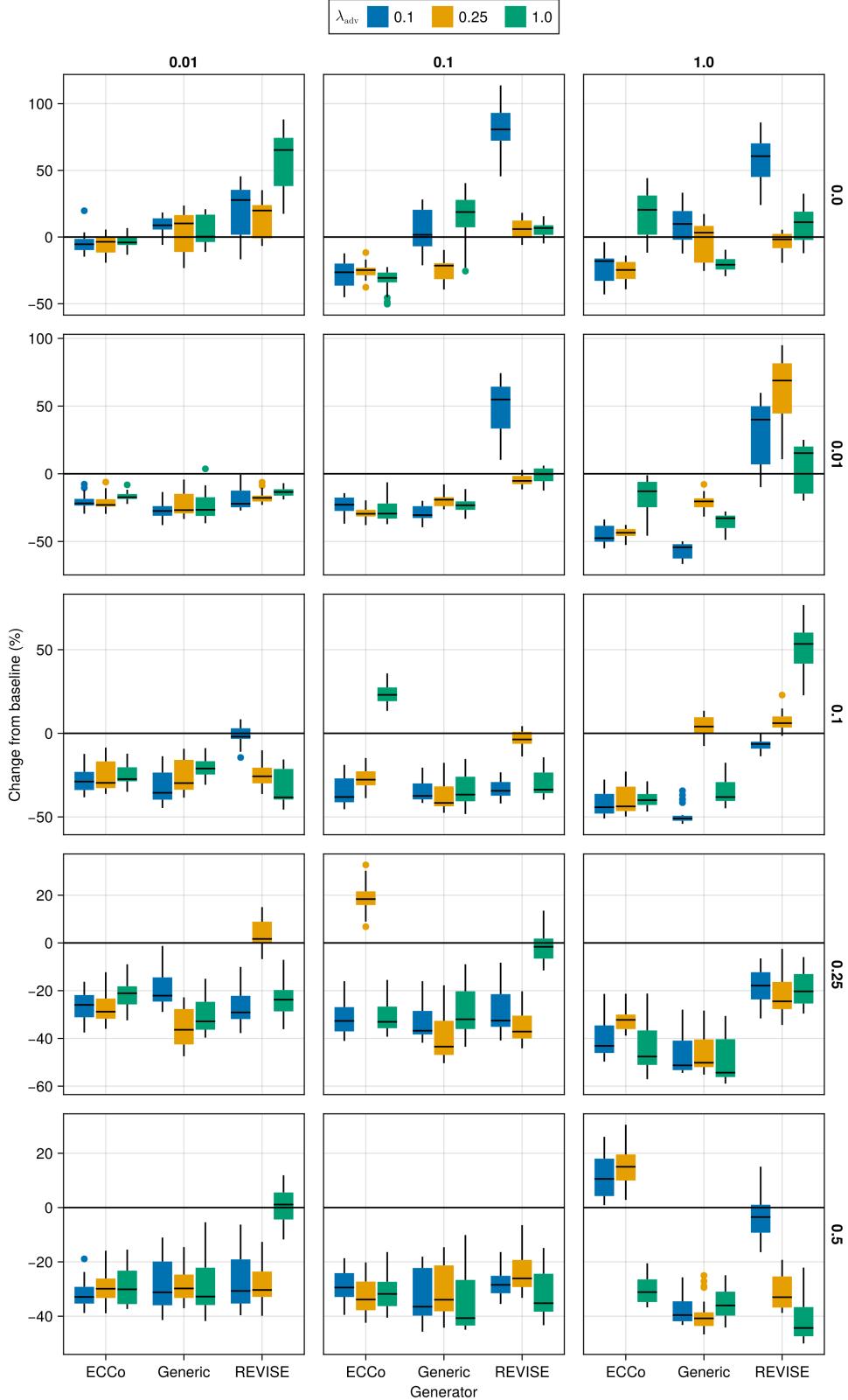


Figure 18: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

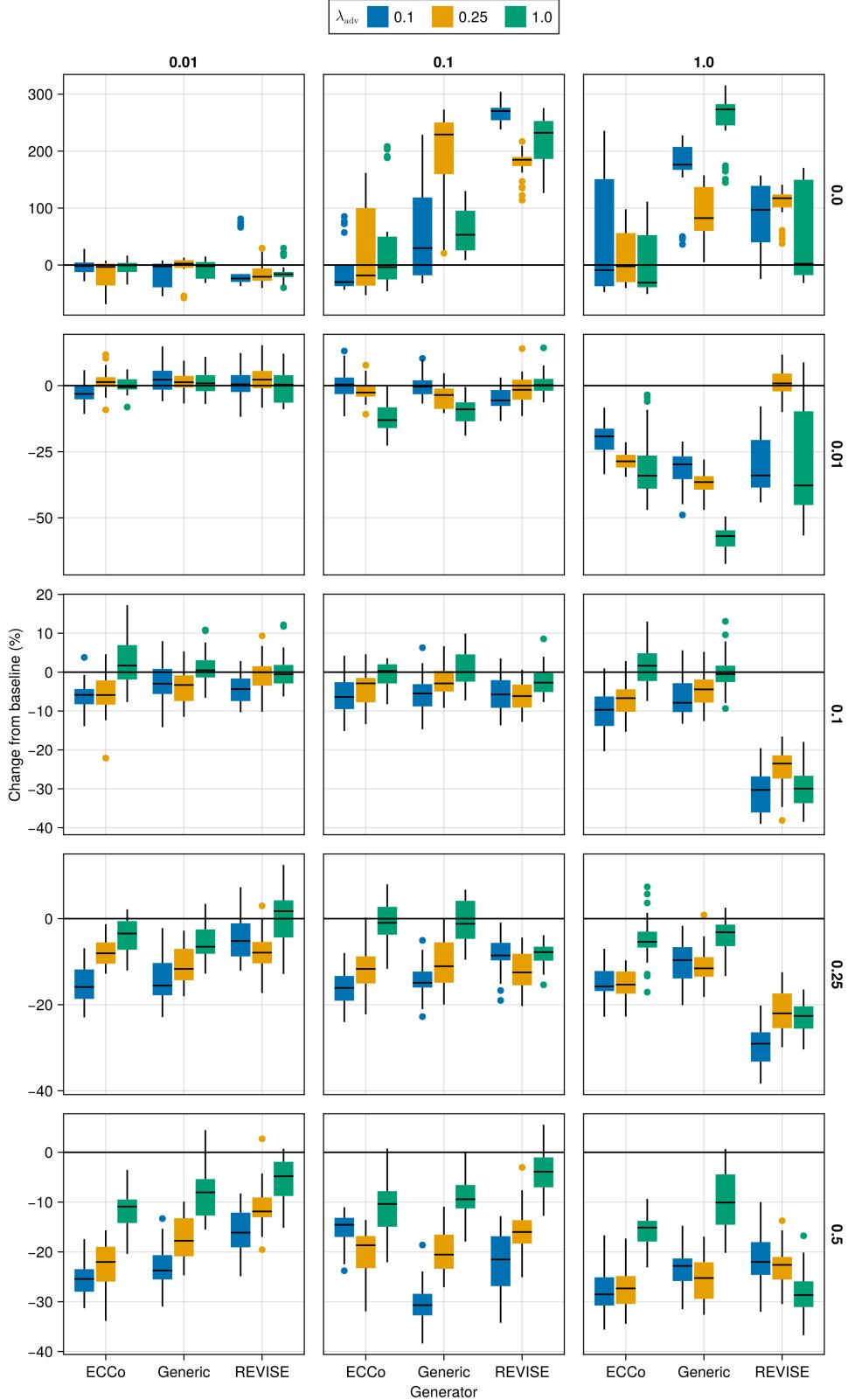


Figure 19: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

Table 5: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 7) and evaluation-phase parameters (Note 8).

Dataset	Variable	Objective	Mean	Std
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
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

<sup>715</sup> **D.4.2 Plausibility**

<sup>716</sup> The results with respect to the plausibility measure are shown in Figure 20 to Figure 23.

<sup>717</sup> **D.4.3 Cost**

<sup>718</sup> The results with respect to the cost measure are shown in Figure 24 to Figure 27.

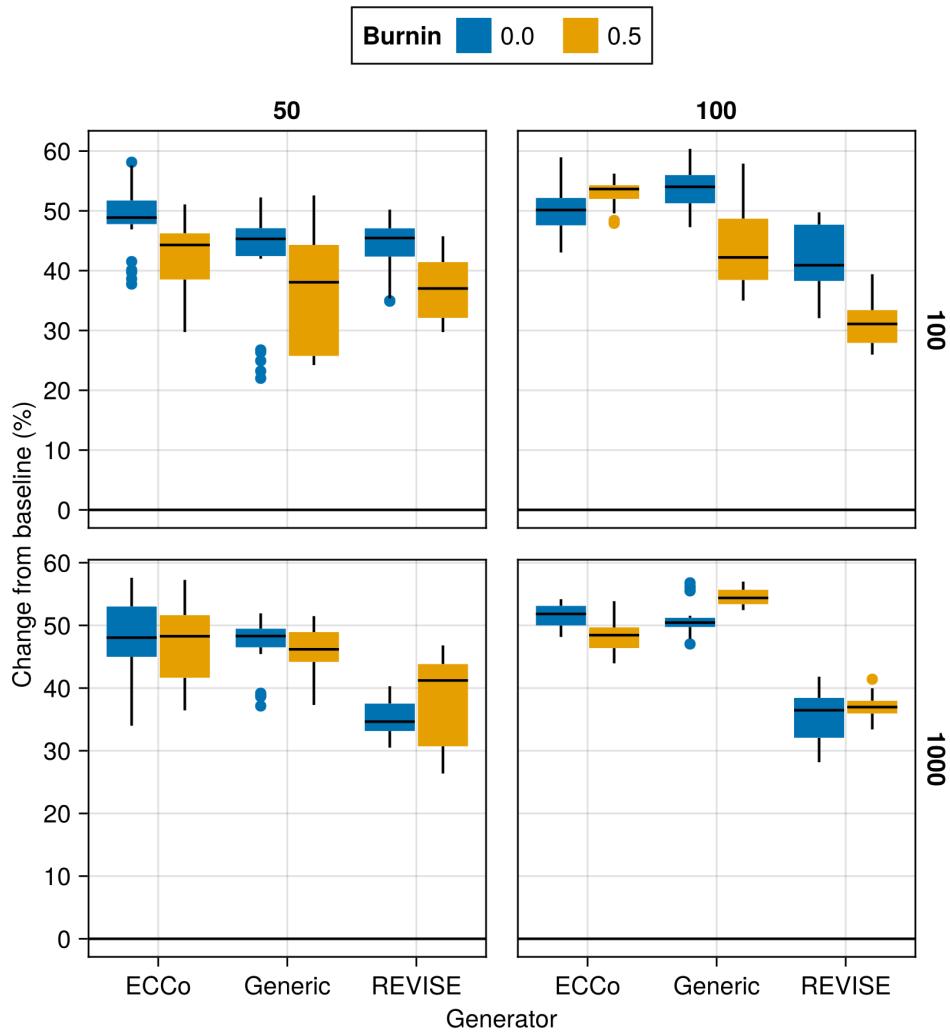


Figure 20: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

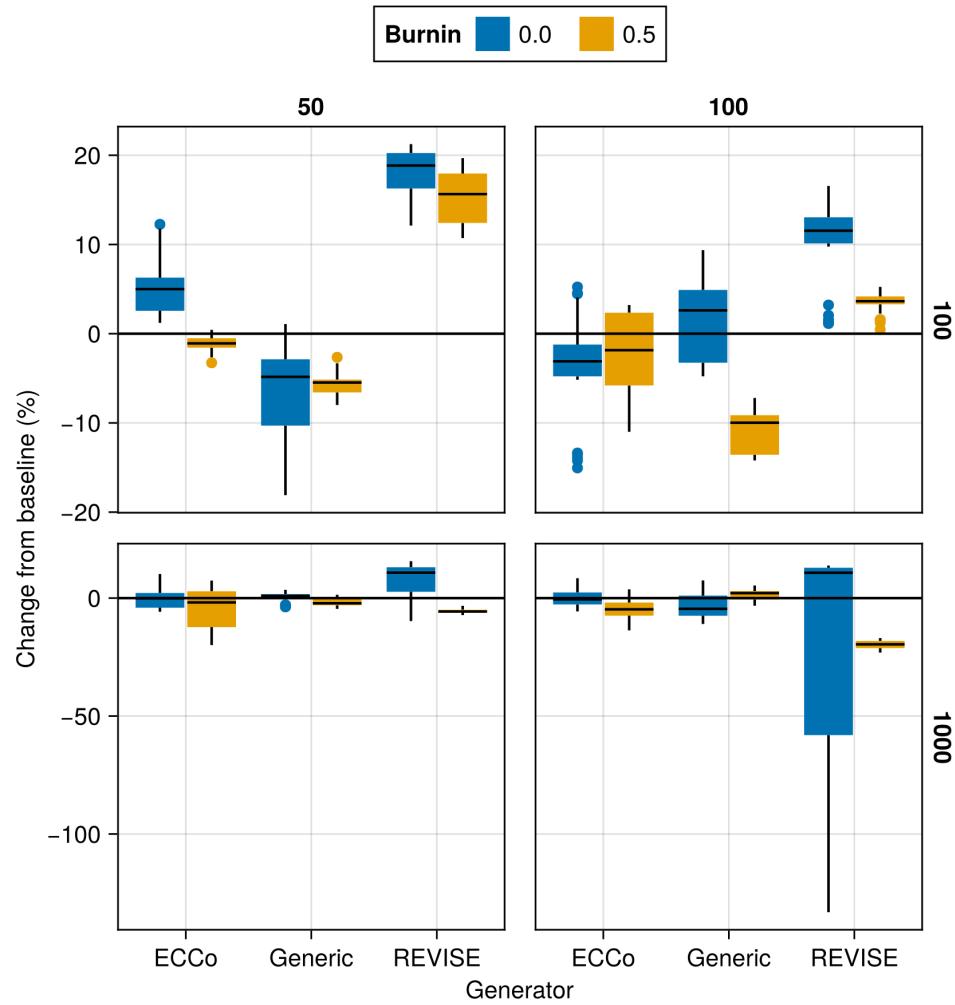


Figure 21: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

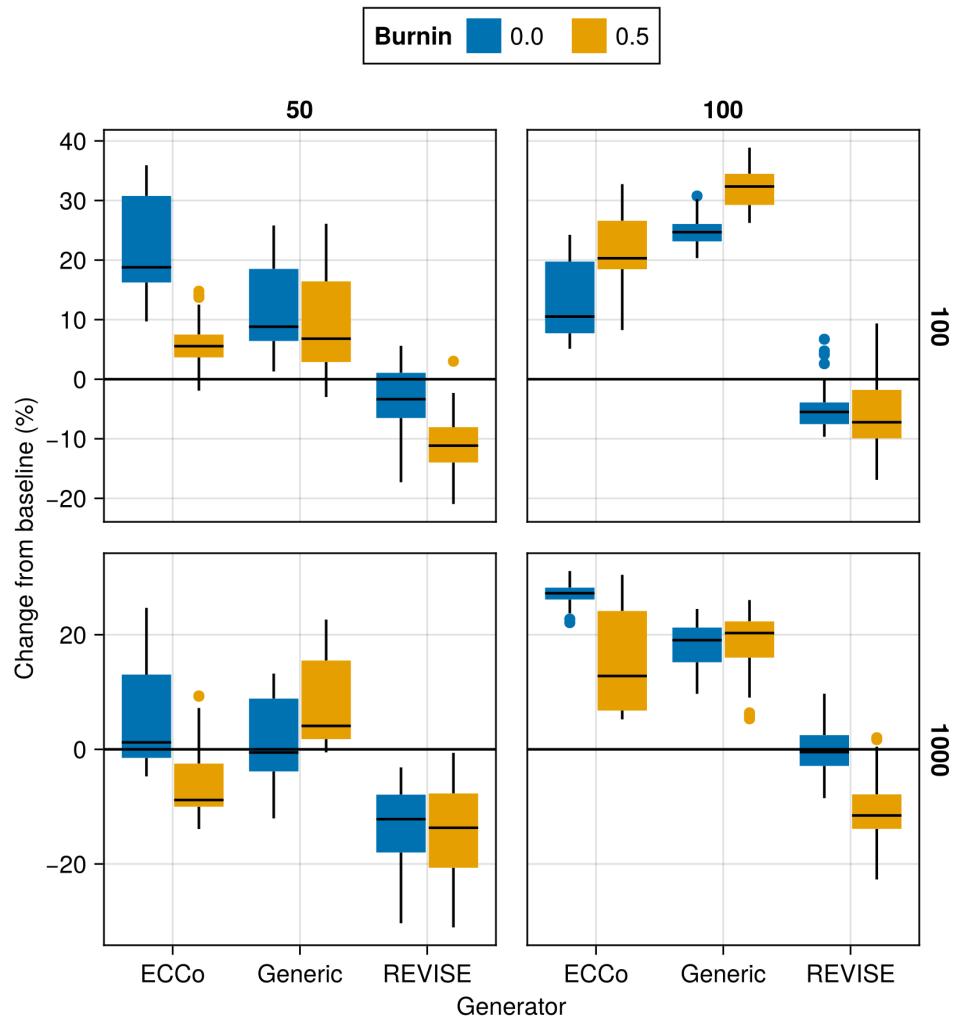


Figure 22: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

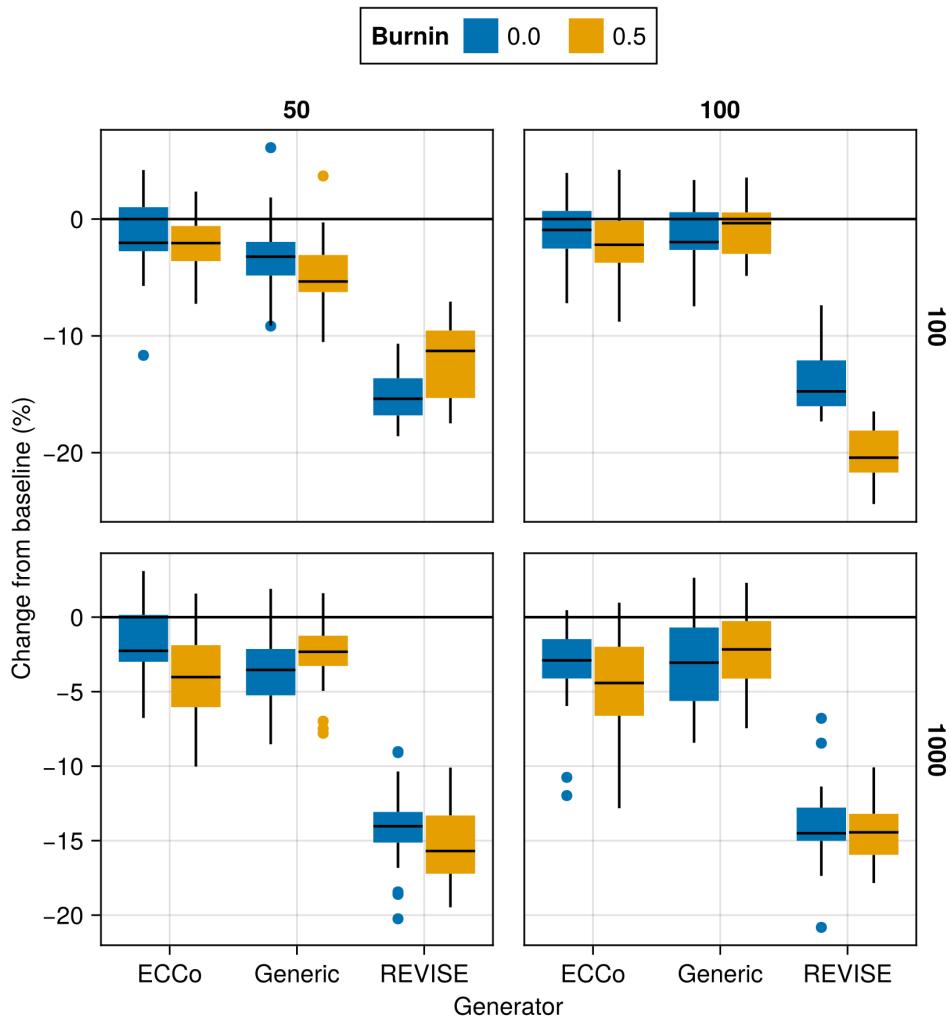


Figure 23: Average outcomes for the plausibility measure across hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

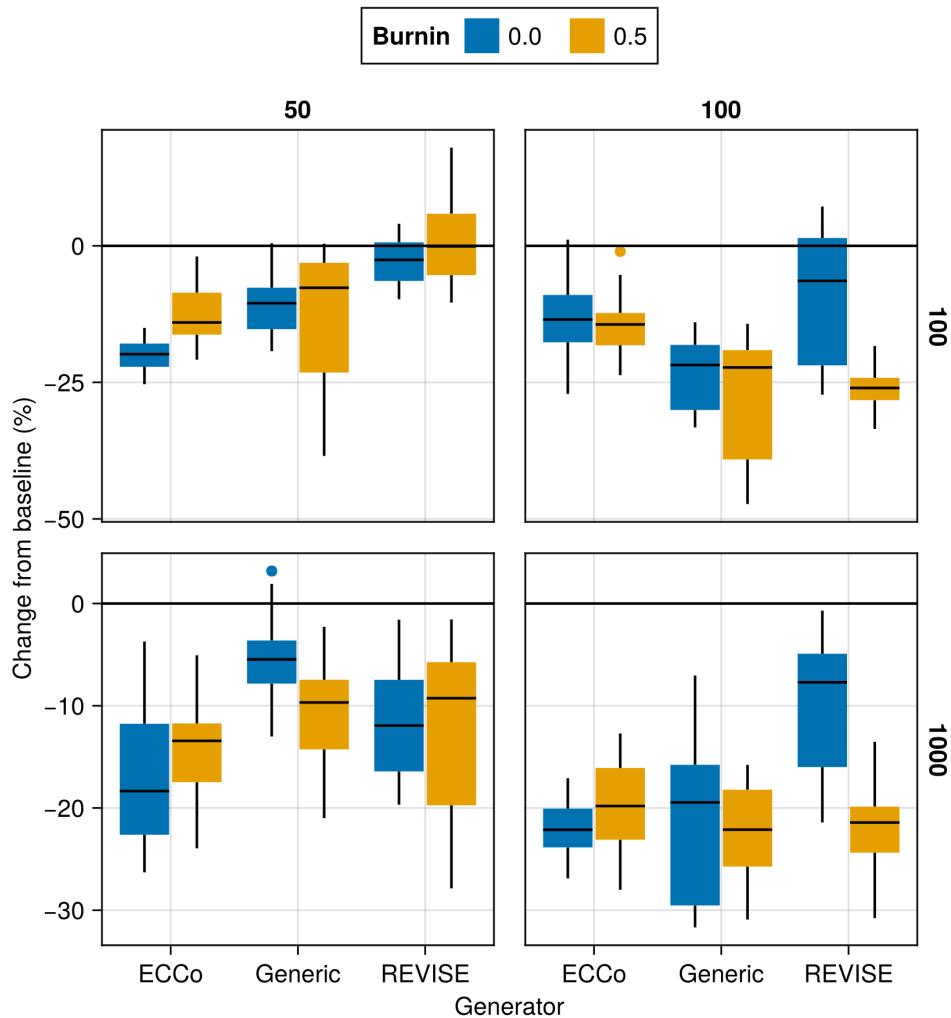


Figure 24: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

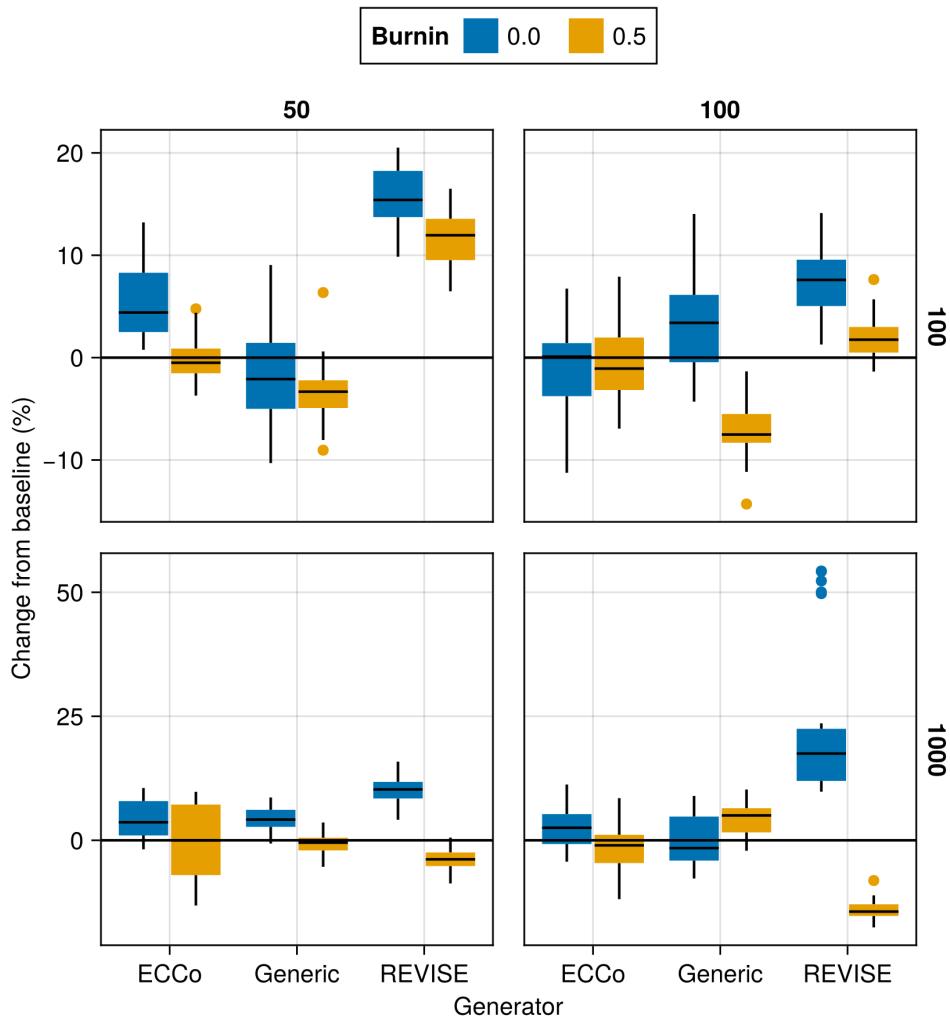


Figure 25: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

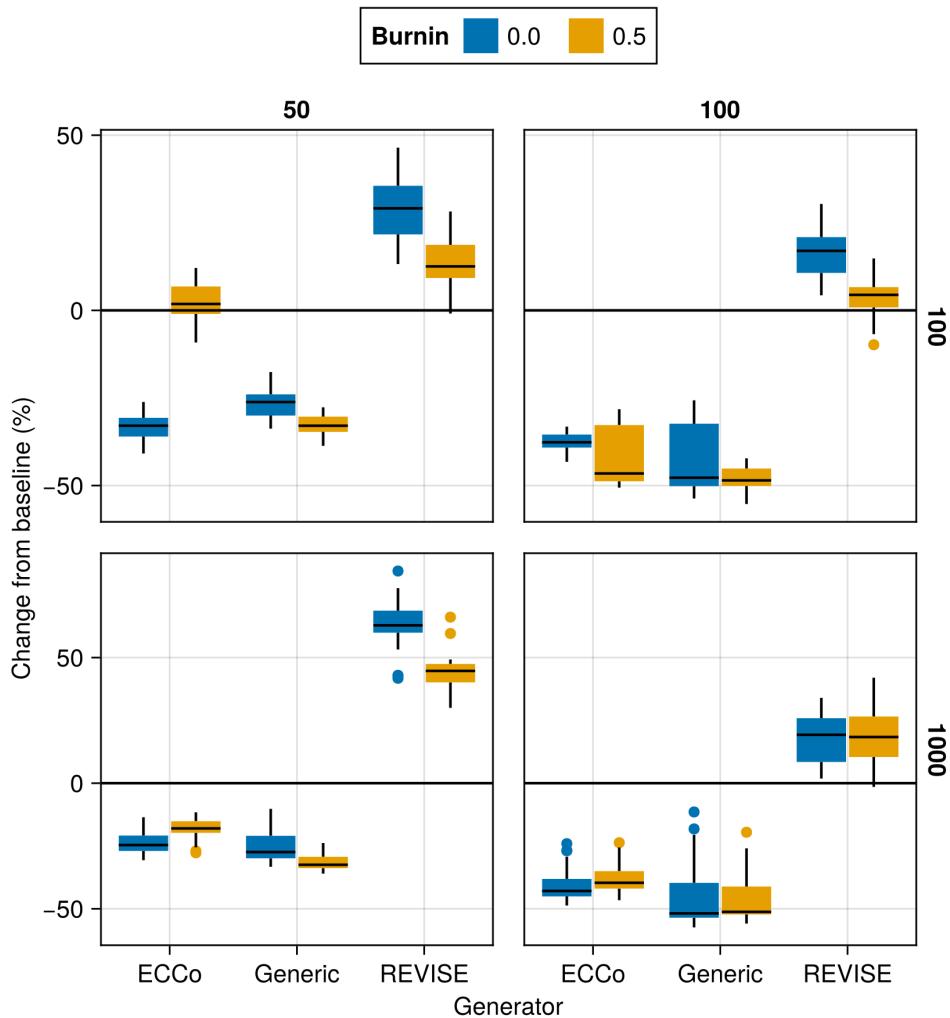


Figure 26: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

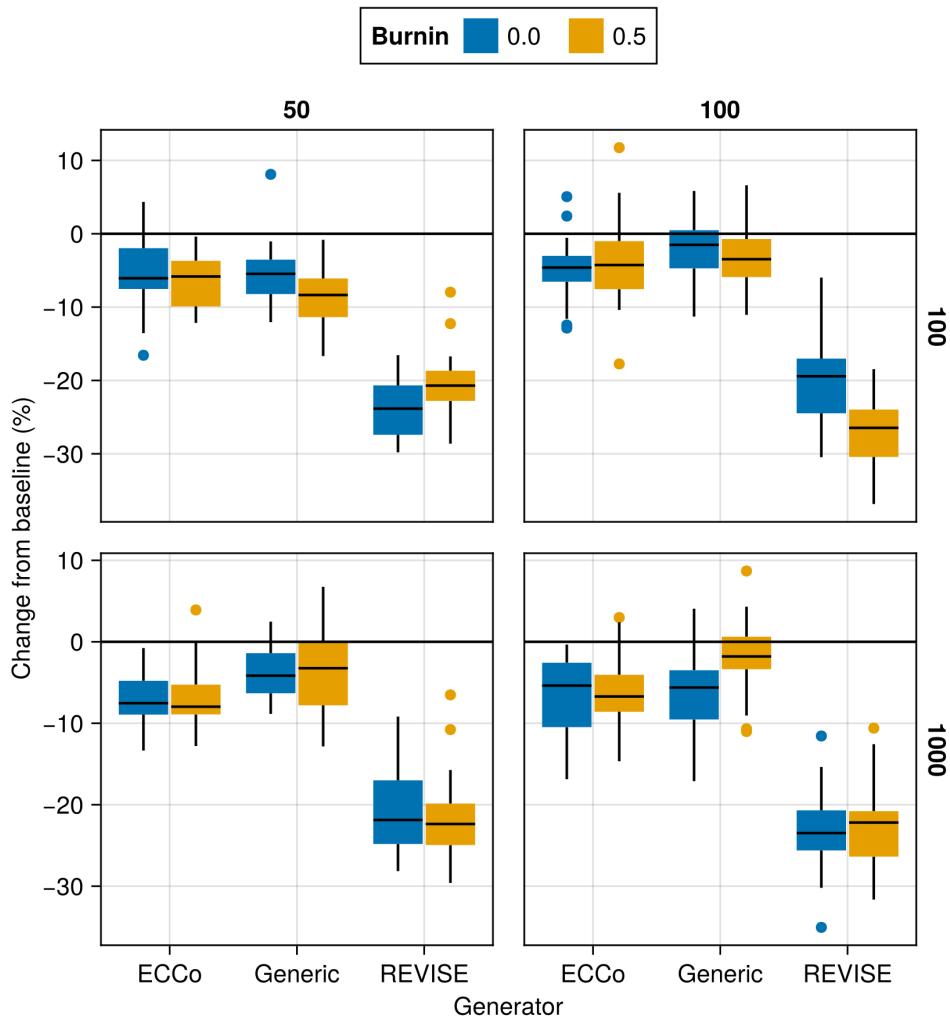


Figure 27: Average outcomes for the cost measure across hyperparameters. This shows the % change from the baseline model for the distance-based cost metric ([Wachter, Mittelstadt, and Russell 2017](#)). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*).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

719 **Appendix E Tuning Key Parameters**

720 Based on the findings from our initial large grid searches (Section D), we tune selected hyperparameters for all datasets:  
 721 namely, the decision threshold  $\tau$  and the strength of the energy regularization  $\lambda_{\text{reg}}$ . The final hyperparameter choices  
 722 for each dataset are presented in Table 2 in Section C. Detailed results for each data set are shown in Figure 28 to  
 723 Figure 45. From Table 2, we notice that the same decision threshold of  $\tau = 0.5$  is optimal for all but one dataset. We  
 724 attribute this to the fact that a low decision threshold results in a higher share of mature counterfactuals and hence more  
 725 opportunities for the model to learn from examples (Figure 37 to Figure 45). This has played a role in particular for  
 726 our real-world tabular datasets and MNIST, which suffered from low levels of maturity for higher decision thresholds.  
 727 In cases where maturity is not an issue, as for *Moons*, higher decision thresholds lead to better outcomes, which may  
 728 have to do with the fact that the resulting counterfactuals are more faithful to the model. Concerning the regularization  
 729 strength, we find somewhat high variation across datasets. Most notably, we find that relatively low levels of regulariza-  
 730 tion are optimal for MNIST. We hypothesize that this finding may be attributed to the uniform scaling of all input  
 731 features (digits).

732 Finally, to increase the proportion of mature counterfactuals for some datasets, we have also investigated the effect on  
 733 the learning rate  $\eta$  for the counterfactual search and even smaller regularization strengths for a fixed decision threshold  
 734 of 0.5 (Figure 46 to Figure 51). For the given low decision threshold, we find that the learning rate has no discernable  
 735 impact on the proportion of mature counterfactuals (Figure 52 to Figure 57). We do notice, however, that the results  
 736 for MNIST are much improved when using a low value  $\lambda_{\text{reg}}$ , the strength for the energy regularization: plausibility is  
 737 increased by up to ~10% (Figure 50) and the proportion of mature counterfactuals reaches 100%.

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

Package Version (Reproducibility)

Tuning was run using v1.1.3 of TaijaData. The follow-up version v1.1.4 introduced an option to split real-world tabular datasets into train and test set, ensuring that pre-processing steps like standardization is fit on the training set only. If you are rerunning the tuning experiments with a version of TaijaData that is higher than v1.1.3, than for the default parameters specified in the configuration files, you may end up with slightly different results, although we would not expect any changes in terms of qualitative findings. For exact reproducibility, please use v1.1.3.

740

741 **E.1 Key Parameters**

742 The hyperparameter grid for tuning key parameters is shown in Note 9. The corresponding evaluation grid used for  
 743 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:
  - $\lambda_{\text{reg}}$ : 0.1, 0.25, 0.5
  - Objective: full, vanilla

744

Note 10: Evaluation Phase

- Generator Parameters:
  - $\lambda_{\text{egy}}$ : 0.1, 0.5, 1.0, 5.0, 10.0

745

746 **E.1.1 Plausibility**

747 The results with respect to the plausibility measure are shown in Figure 28 to Figure 36.

748 **E.1.2 Proportion of Mature CE**

749 The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure 37 to Figure 45.

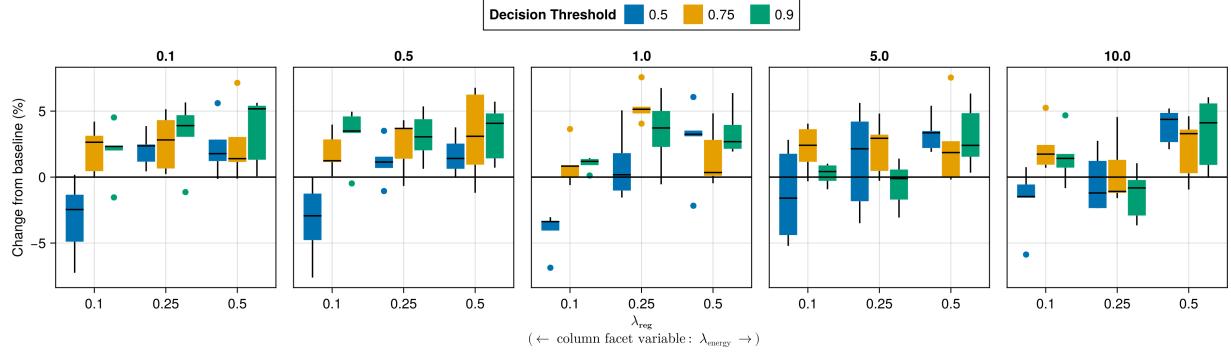


Figure 28: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Adult.

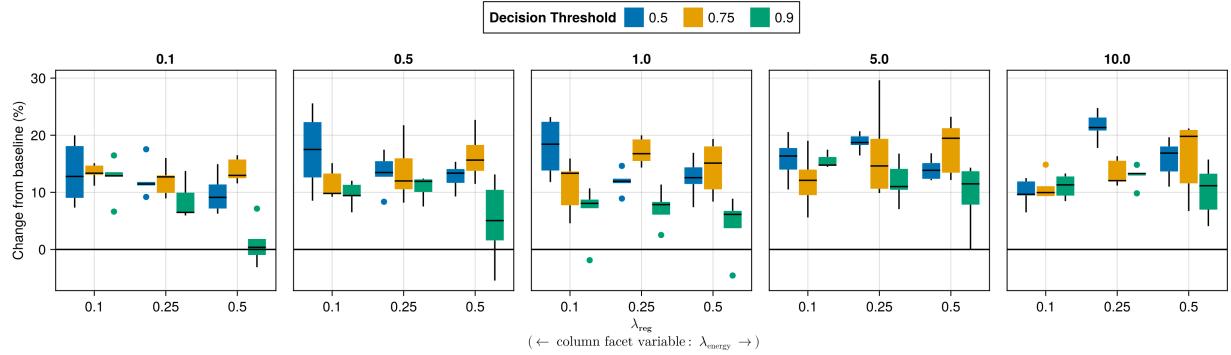


Figure 29: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: California Housing.

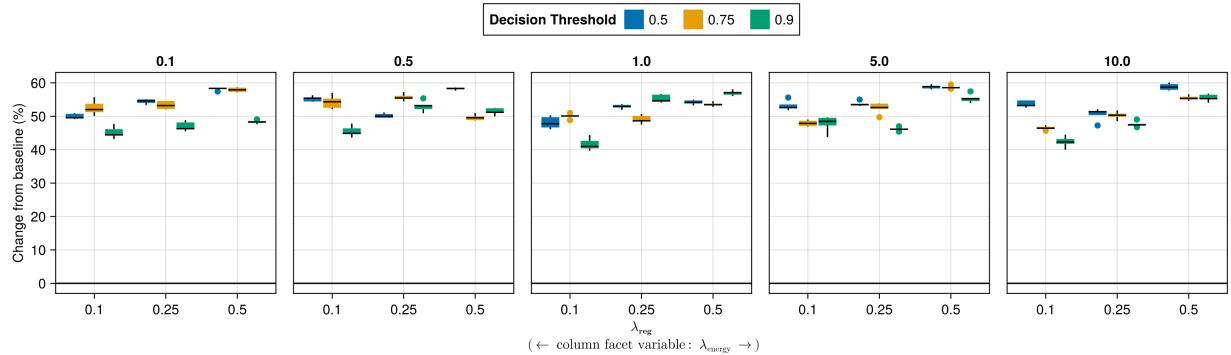


Figure 30: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Circles.

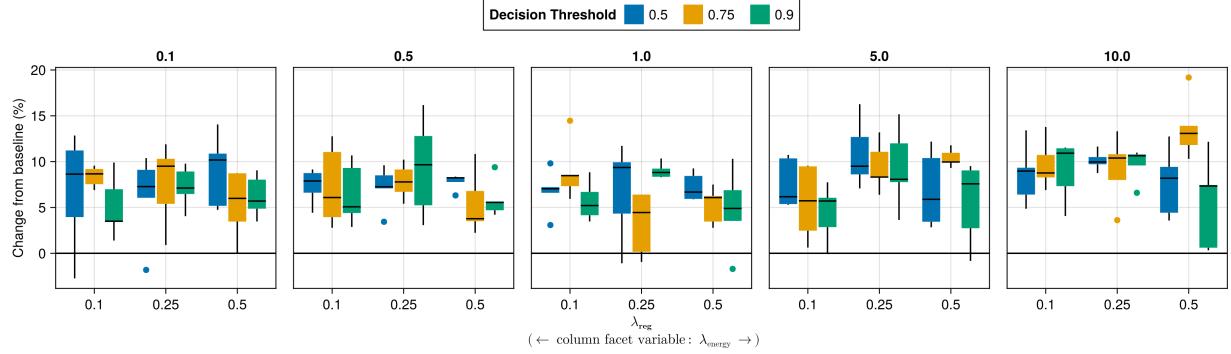


Figure 31: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Credit.

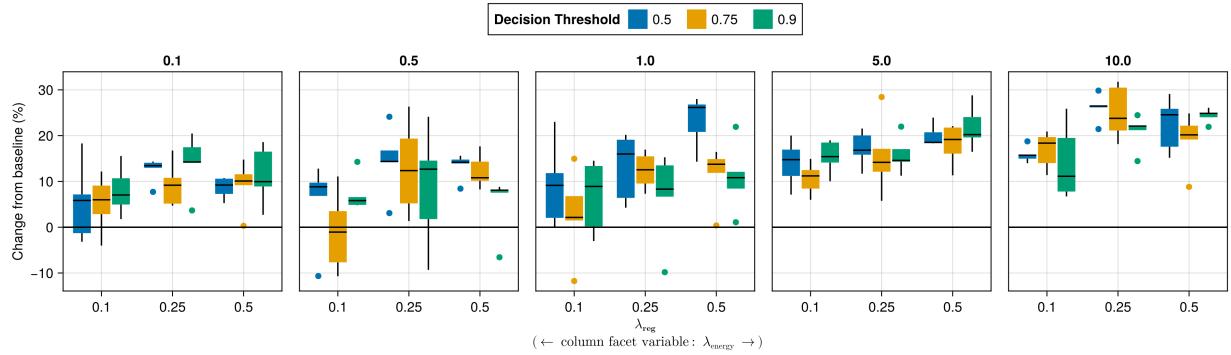


Figure 32: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: GMSC.

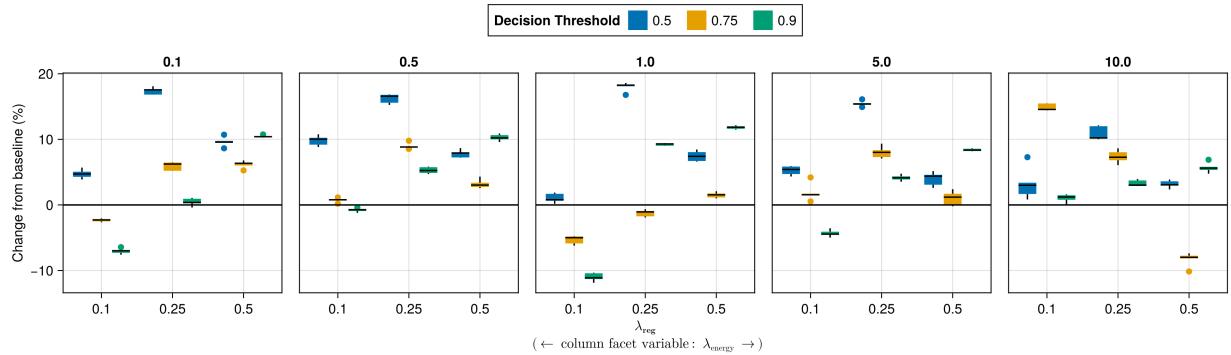


Figure 33: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Linearly Separable.

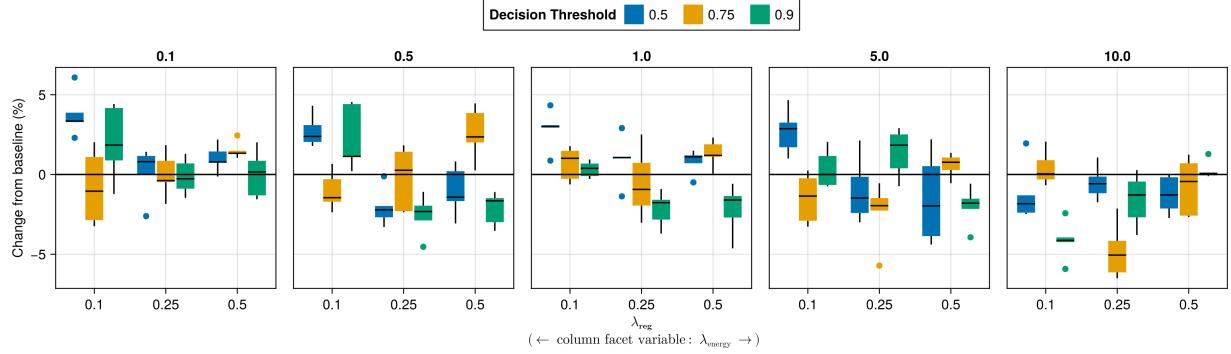


Figure 34: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: MNIST.

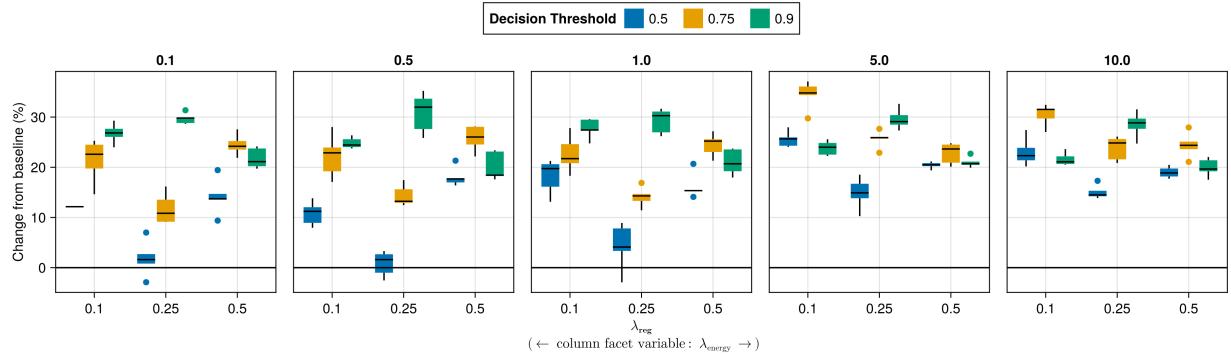


Figure 35: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Moons.

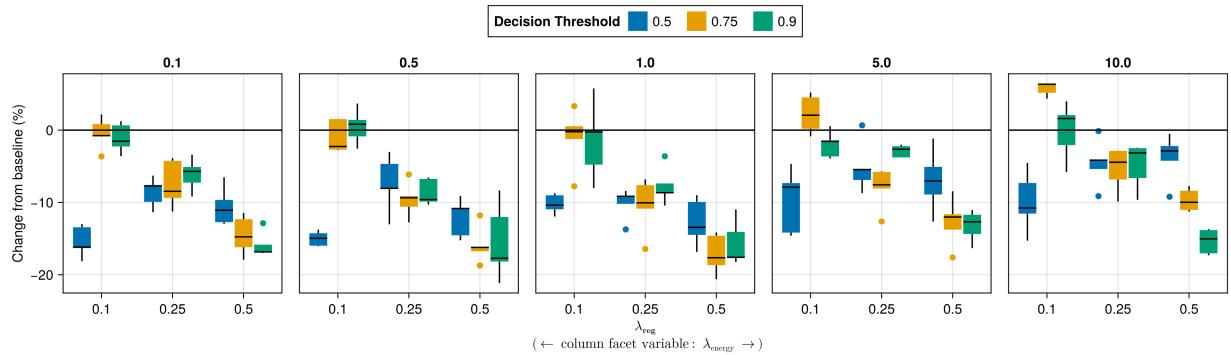


Figure 36: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

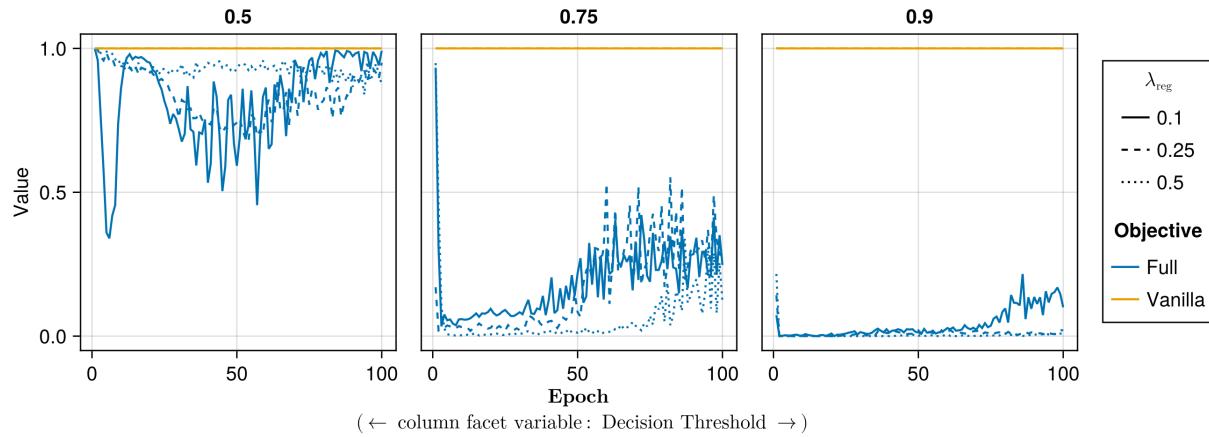


Figure 37: Proportion of mature counterfactuals in each epoch. Data: Adult.

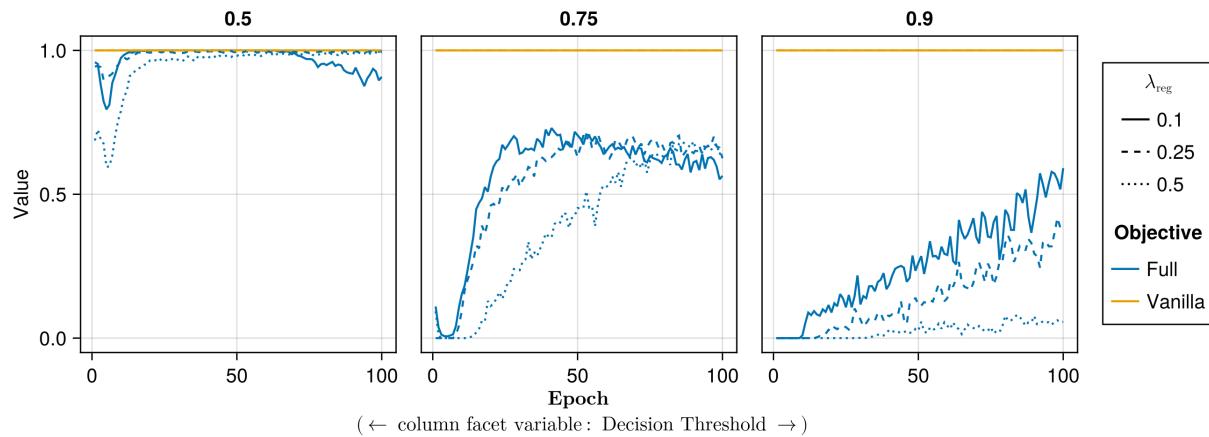


Figure 38: Proportion of mature counterfactuals in each epoch. Data: California Housing.

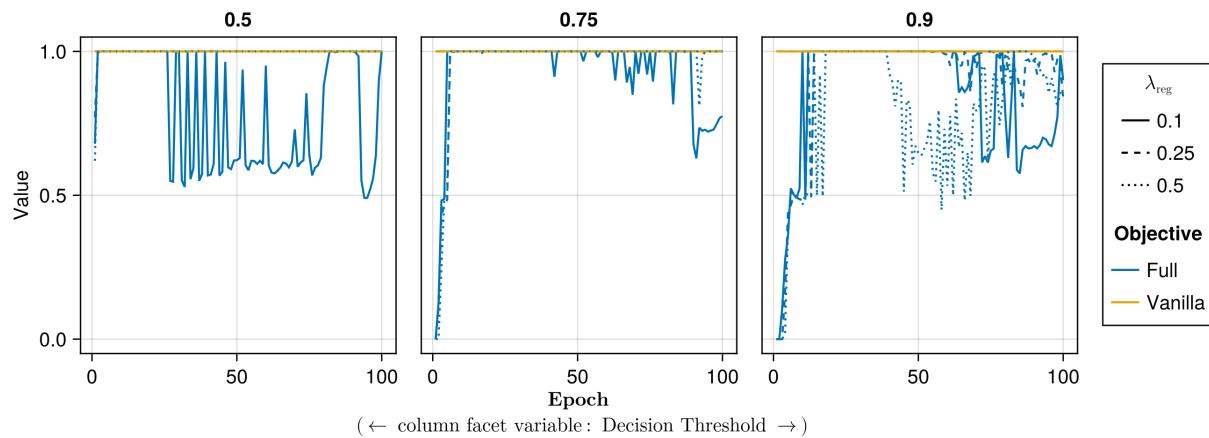


Figure 39: Proportion of mature counterfactuals in each epoch. Data: Circles.

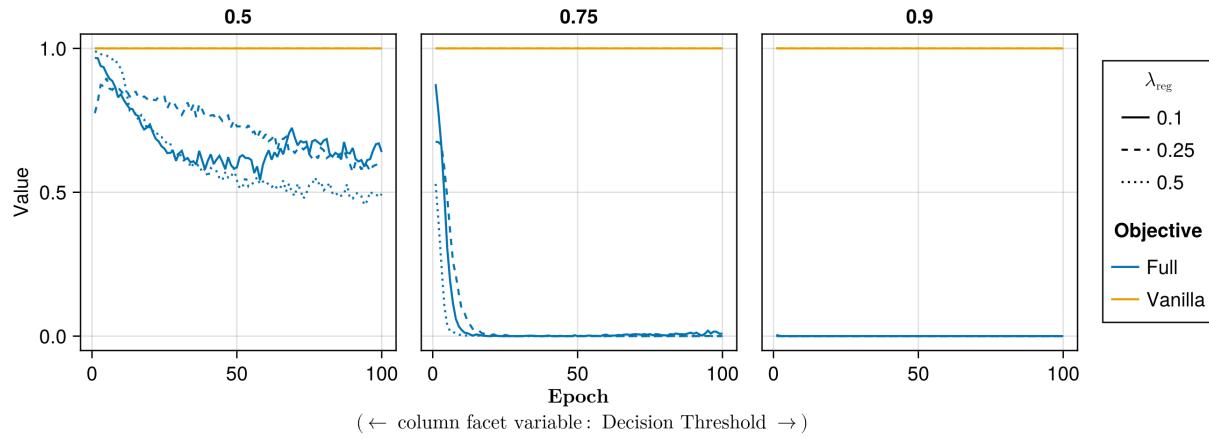


Figure 40: Proportion of mature counterfactuals in each epoch. Data: Credit.

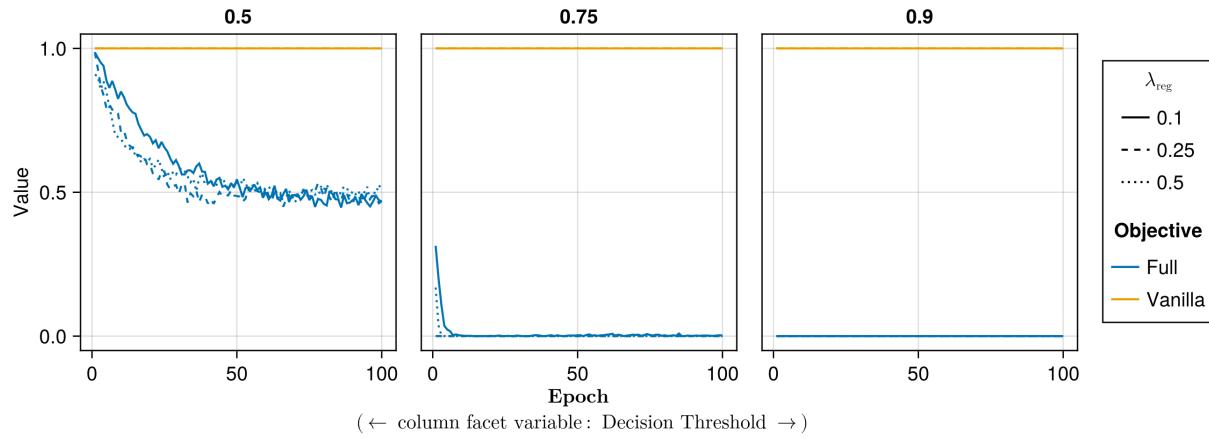


Figure 41: Proportion of mature counterfactuals in each epoch. Data: GMSC.

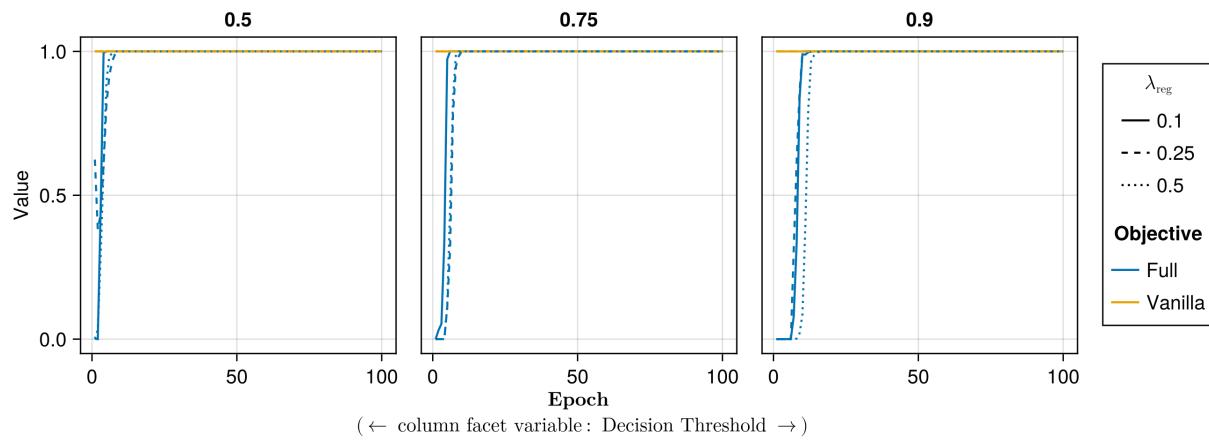


Figure 42: Proportion of mature counterfactuals in each epoch. Data: Linearly Separable.

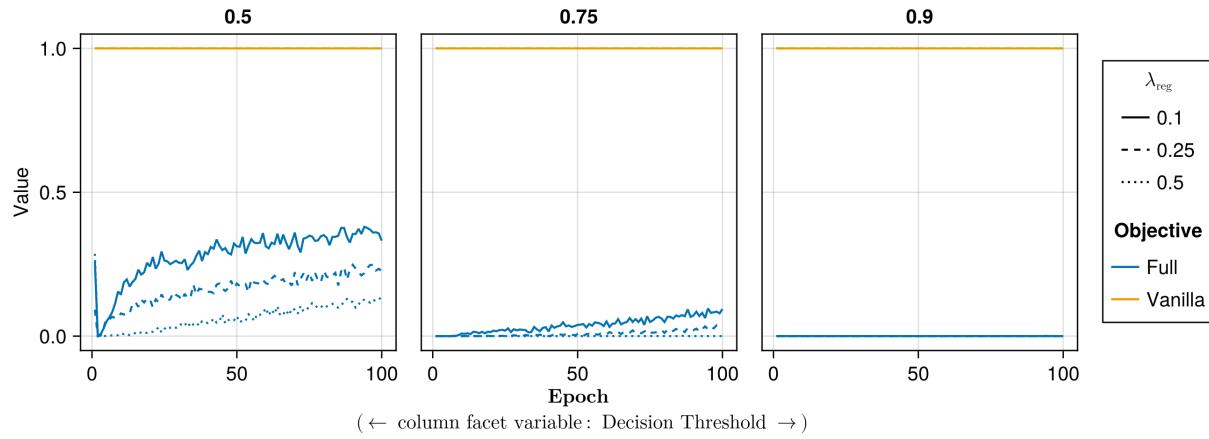


Figure 43: Proportion of mature counterfactuals in each epoch. Data: MNIST.

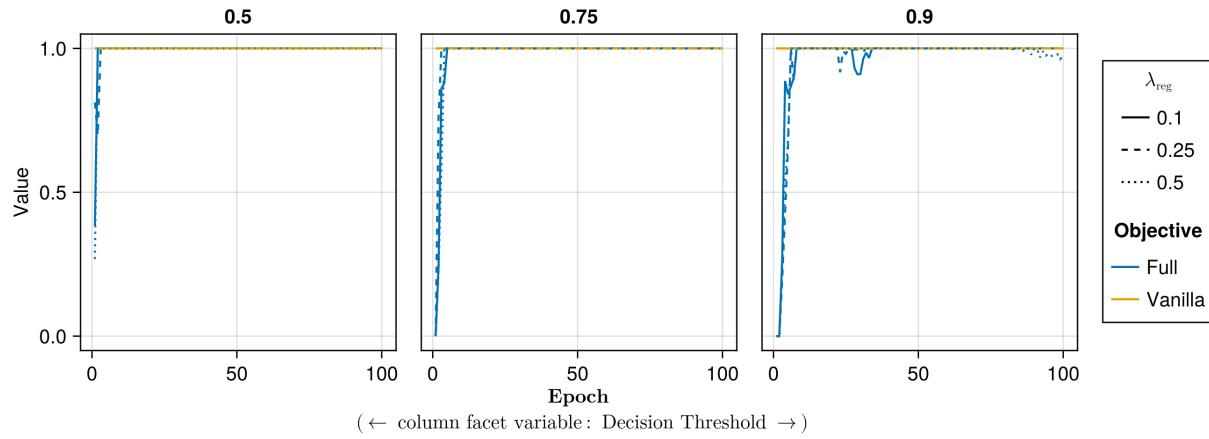


Figure 44: Proportion of mature counterfactuals in each epoch. Data: Moons.

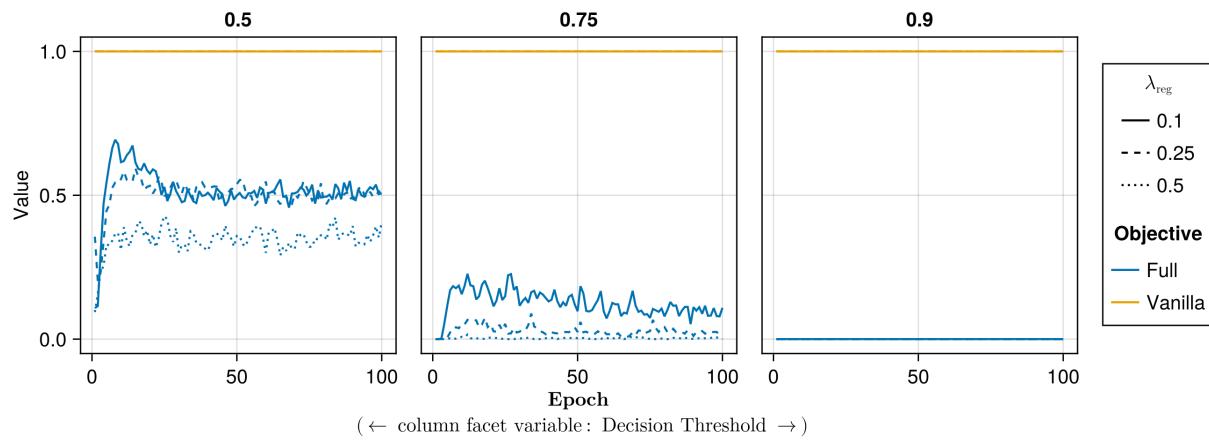


Figure 45: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

750 **E.2 Learning Rate**

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

Note 11: Training Phase

- Generator Parameters:
  - Learning Rate: 0.1, 0.5, 1.0
- Model: mlp
- Training Parameters:
  - $\lambda_{\text{reg}}$ : 0.01, 0.1, 0.5
  - Objective: full, vanilla

753

Note 12: Evaluation Phase

- Generator Parameters:
  - $\lambda_{\text{egy}}$ : 0.1, 0.5, 1.0, 5.0, 10.0

754

755 **E.2.1 Plausibility**

756 The results with respect to the plausibility measure are shown in Figure 46 to Figure 51.

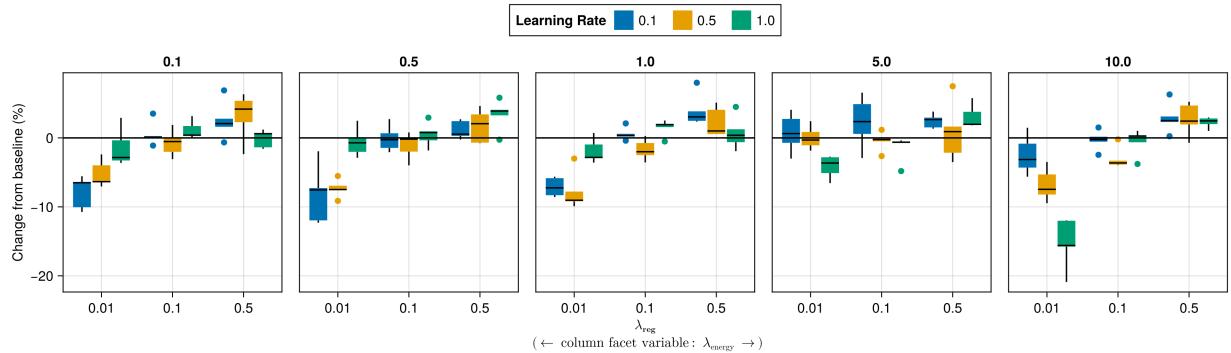


Figure 46: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Adult.

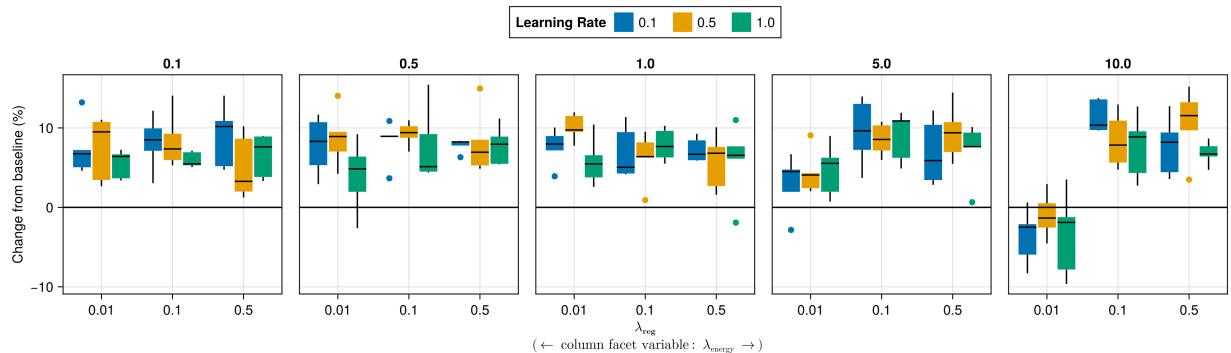


Figure 47: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Credit.

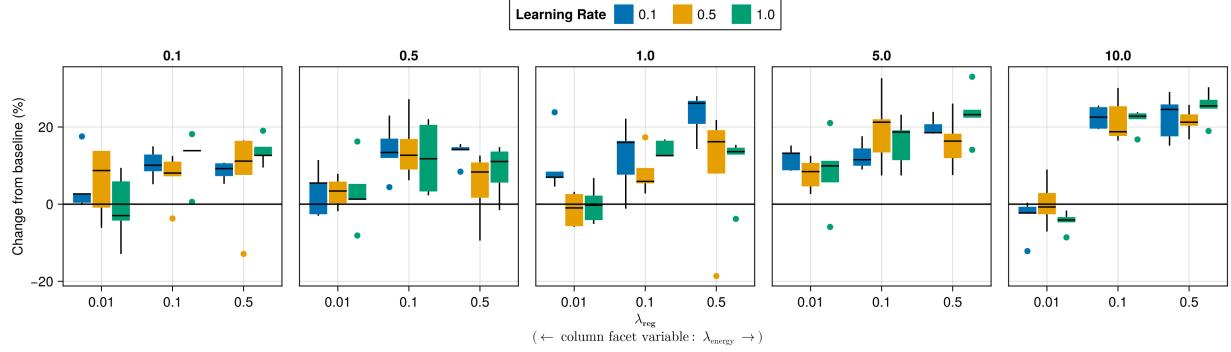


Figure 48: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for  $ECCo$ ). Data: GMSC.

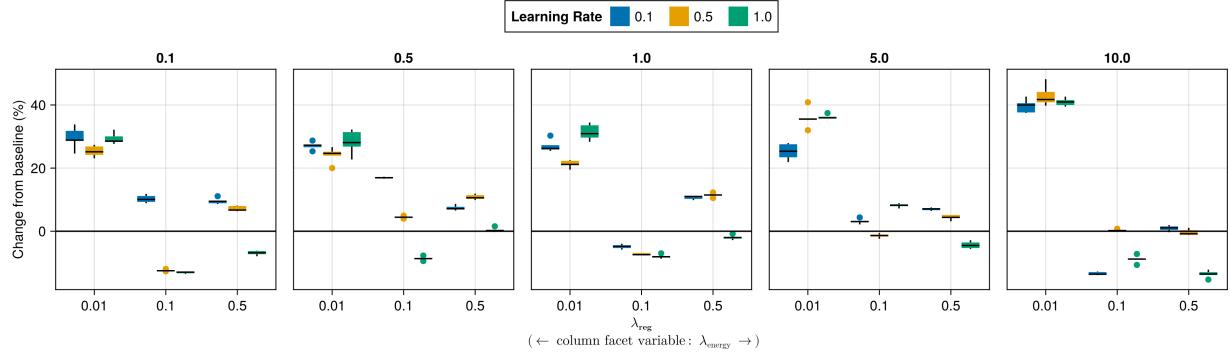


Figure 49: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for  $ECCo$ ). Data: Linearly Separable.

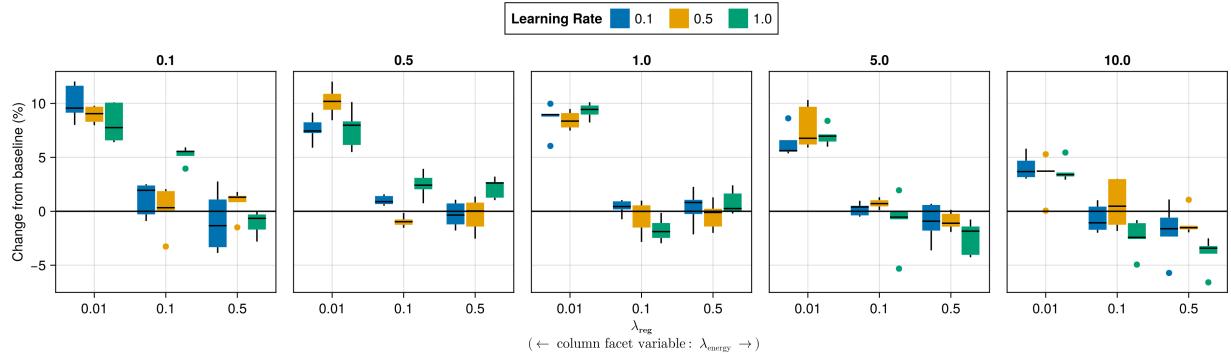


Figure 50: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for  $ECCo$ ). Data: MNIST.

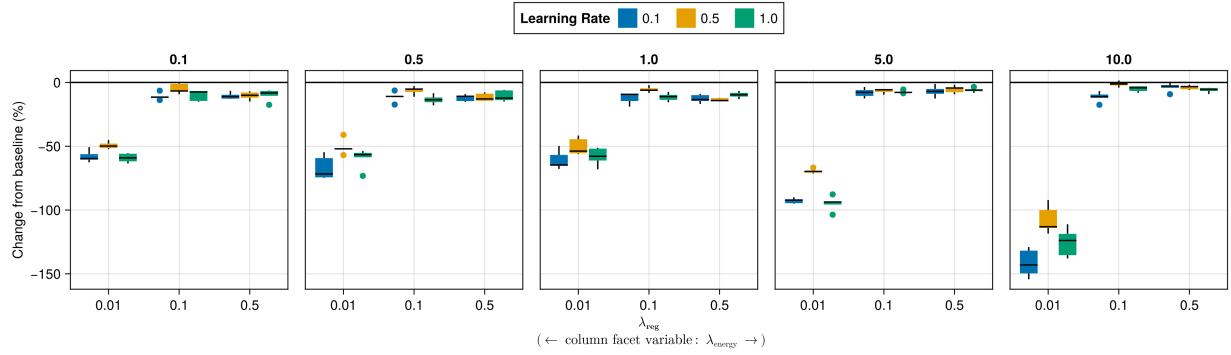


Figure 51: Average outcomes for the plausibility measure across key hyperparameters. This shows the % change from the baseline model for the distance-based implausibility metric (Equation 4). Boxplots indicate the variation across evaluation runs and test settings (varying parameters for *ECCo*). Data: Overlapping.

### 757 E.2.2 Proportion of Mature CE

758 The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure 52 to Figure 57.

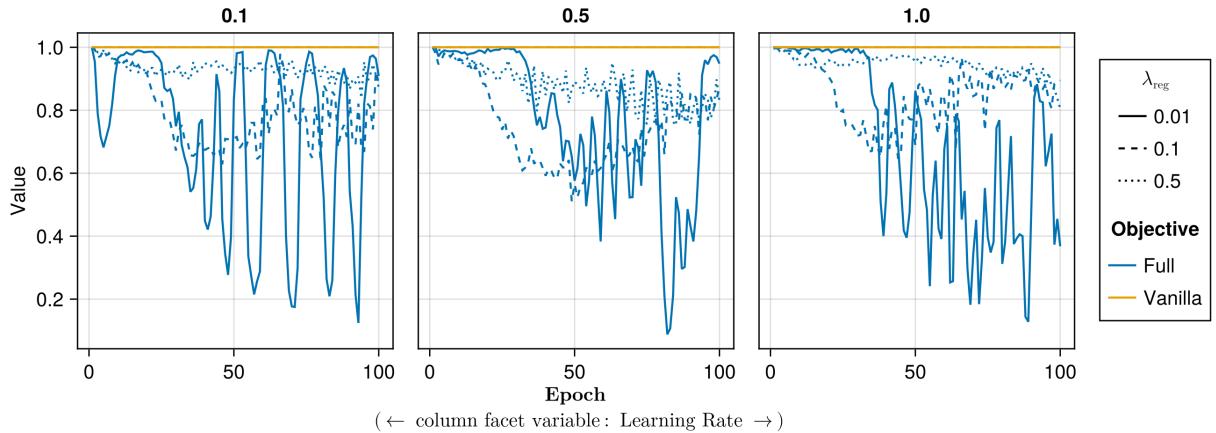


Figure 52: Proportion of mature counterfactuals in each epoch. Data: Adult.

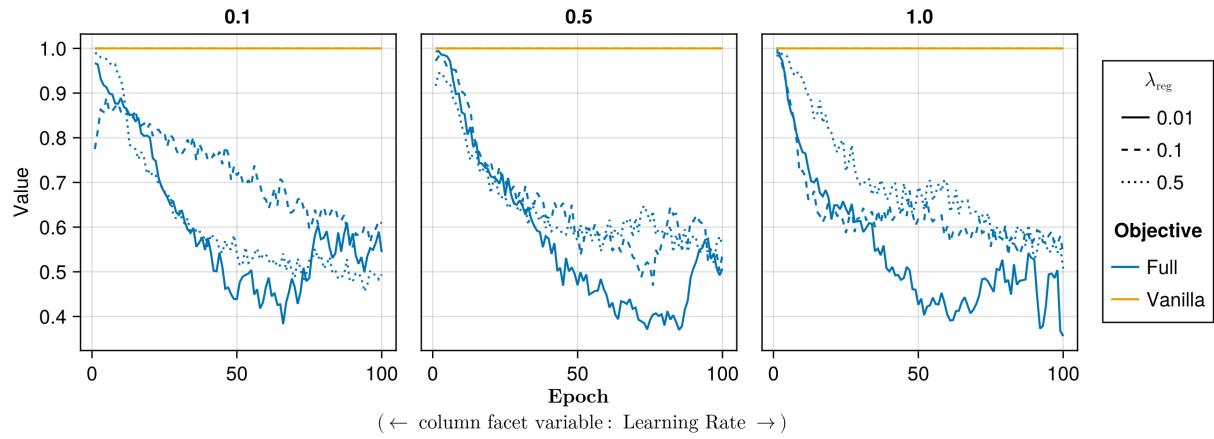


Figure 53: Proportion of mature counterfactuals in each epoch. Data: Credit.

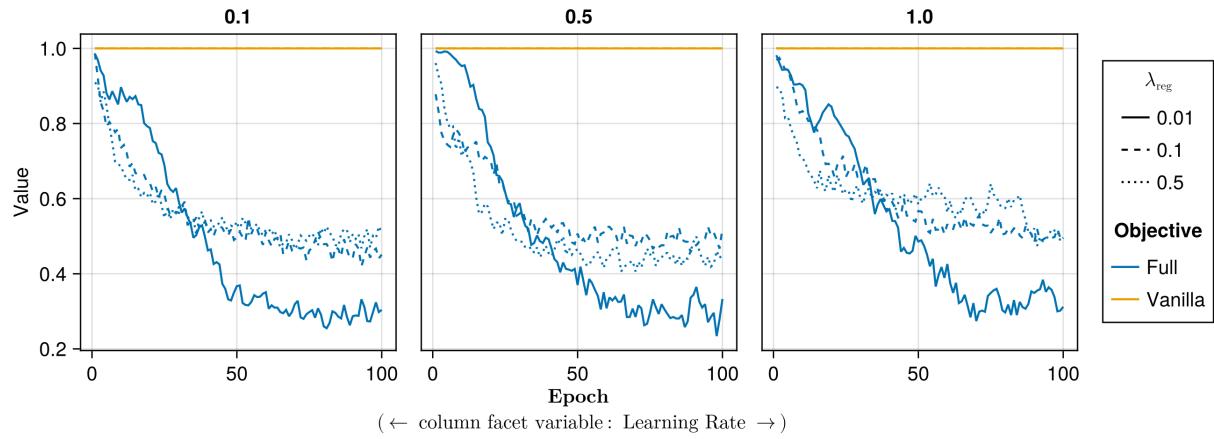


Figure 54: Proportion of mature counterfactuals in each epoch. Data: GMSC.

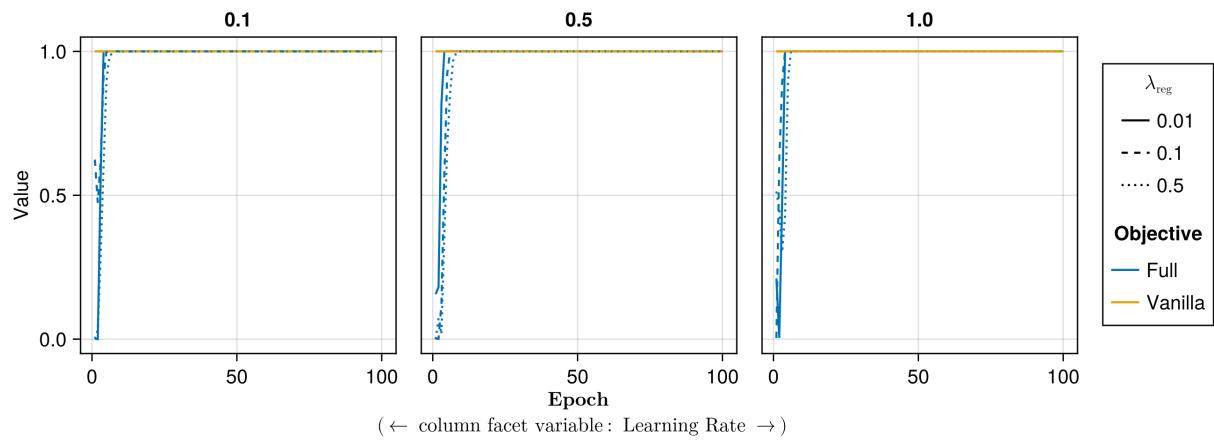


Figure 55: Proportion of mature counterfactuals in each epoch. Data: Linearly Separable.

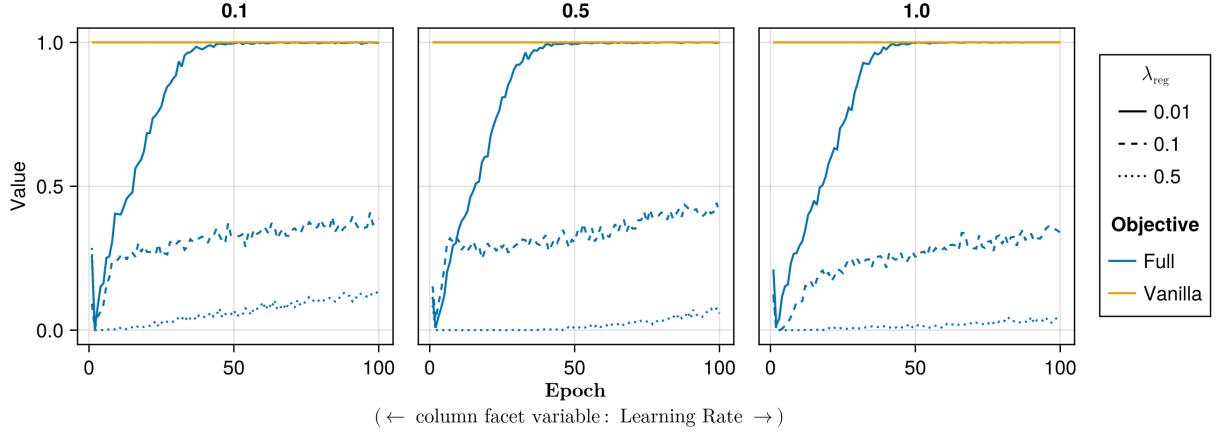


Figure 56: Proportion of mature counterfactuals in each epoch. Data: MNIST.

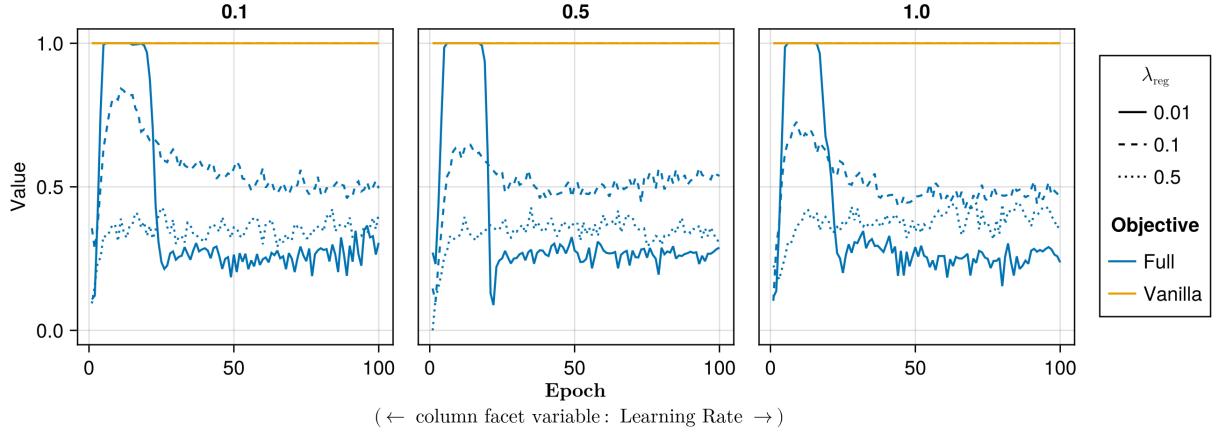


Figure 57: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

## 759 Appendix F Computation Details

### 760 F.1 Hardware

761 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.

### 765 F.1.1 Grid Searches

766 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*), 767 1h4m (*Linearly Separable*), 1h49m (*Circles*), 3h52m (*Overlapping*). Model evaluations for large grid searches were 768 parallelized across 20 CPUs with 3GB memory each. Evaluations for all data sets took less than one hour (<1h) to 769 complete. 770

### 771 F.1.2 Tuning

772 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 773 in less than half an hour (<0h30m). The *Adult* dataset took around 0h35m to complete. Evaluations across 20 CPUs 774 with 3GB memory each generally took less than 0h30m to complete. For *MNIST*, we relied on 100 CPUs with 2GB 775 memory each. For the *MLP*, training of all models could be completed in 1h30m, while the evaluation across 20 CPUs 776

777 (6GB memory) took 4h12m. For the *CNN*, training of all models took ~8h, with conventionally trained models taking  
778 ~0h15m each and model with CT taking ~0h30m-0h45m each.

779 **F.2 Software**

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