
COUNTERFACTUAL TRAINING: TEACHING MODELS PLAUSIBLE AND ACTIONABLE EXPLANATIONS

A PREPRINT

Patrick Altmeyer 

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

p.altmeyer@tudelft.nl

Aleksander Buszydlik

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

Arie van Deursen

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

Cynthia C. S. Liem

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

March 13, 2025

ABSTRACT

We propose a novel training regime called Counterfactual Training that leverages counterfactual explanations to increase the explanatory capacity of models. Counterfactual explanations have emerged as a popular post-hoc explanation method for opaque machine learning models: they inform how factual inputs would need to change in order for a model to produce some desired output. To be useful in real-word decision-making systems, counterfactuals should be plausible with respect to the underlying data and actionable with respect to stakeholder requirements. 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 this 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 advances in **representation learning**: instead of relying on features and rules that are carefully hand-crafted by humans, modern machine learning (ML)

- 18 models are tasked with learning representations directly from data, guided by narrow objectives such as predictive
 19 accuracy (I. Goodfellow, Bengio, and Courville 2016). Modern advances in computing have made it possible to
 20 provide such models with ever-growing degrees of freedom to achieve that task, which has often led them to outperform
 21 traditionally more parsimonious models. Unfortunately, in doing so, models learn increasingly complex and highly
 22 sensitive representations that we 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 revolution, Szegedy et al. (2013) showed that artificial neural networks (ANN) are
 25 sensitive to adversarial examples: counterfactuals of model inputs that yield vastly different model predictions despite
 26 being “imperceptible” in that they are semantically indifferent from their factual counterparts. Although some partially
 27 effective mitigation strategies have been proposed, for example **adversarial training** (I. J. Goodfellow, Shlens, and
 28 Szegedy 2014), truly robust deep learning (DL) remains unattainable even for models that are considered shallow by
 29 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).¹ 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 and
 33 Carbin 2019) and generalizability (Morcos et al. 2019). Similarly, Zhang et al. (2021) showed that state-of-the-art
 34 neural networks are so expressive that they can fit randomly labeled data. Thus, looking at the predictive performance,
 35 the solutions may seem to provide compelling explanations for the data, when in fact they are based on purely as-
 36 sociative, semantically meaningless patterns. This poses two related challenges. Firstly, there is no dependable way
 37 to verify if such complex representations correspond to meaningful and plausible explanations. Secondly, even if we
 38 could resolve the first challenge, it remains undecided how to ensure that models can *only* learn valuable explanations.
- 39 The first challenge has attracted an abundance of research on **explainable AI** (XAI) which aims to develop tools to
 40 derive explanations from complex model representations. This can mitigate a scenario in which we deploy opaque
 41 models and blindly rely on their predictions. On countless occasions, this scenario has occurred in practice and caused
 42 real harm to people who were affected adversely and often unfairly by automated decision-making (ADM) systems
 43 involving opaque models (O’Neil 2016; McGregor 2021). Effective XAI tools can aid us in monitoring models and
 44 providing recourse to individuals to turn adverse outcomes (e.g., “loan application rejected”) into positive ones (e.g.,
 45 “application accepted”). Wachter, Mittelstadt, and Russell (2017) propose **counterfactual explanations** (CE) as an
 46 effective approach to achieve this goal: CEs explain how factual inputs need to change in order for some fitted model
 47 to produce some desired output, typically involving minimal perturbations.
- 48 To our surprise, the second challenge has not yet attracted any major consolidated research effort. Specifically, there
 49 has been no concerted effort towards improving models’ explanatory capacity, which we will henceforth
 50 simply call “explainability”, defined as the degree to which learned representations correspond to explanations that are
 51 interpretable and deemed **plausible** by humans (see Definition 3.1). Instead, the choice has typically been to improve
 52 the ability of XAI tools to identify the subset explanations that are both plausible and valid for any given model,
 53 independent of whether the learned representations are also compatible with implausible explanations (Altmeyer et
 54 al. 2024). Fortunately, recent findings indicate that explainability can arise as byproduct of regularization techniques
 55 aimed at other objectives such as robustness, generalization, and generative capacity (Schut et al. 2021; Augustin,
 56 Meinke, and Hein 2020; Altmeyer et al. 2024).
- 57 Building on these findings, we introduce **counterfactual training**: a novel training regime explicitly geared towards
 58 aligning model representations with plausible explanations. Our contributions are as follows:
- 59 • We discuss existing related work on improving models and consolidate it through the lens of counterfactual
 explanations (Section 2).
 - 60 • We present our proposed methodological framework that leverages faithful counterfactual explanations during
 the training phase of models to achieve the explainability objective (Section 3).
 - 61 • Through extensive experiments we demonstrate the counterfactual training improve model explainability
 while maintaining high predictive performance. We run ablation studies and grid searches to understand
 how the underlying model components and hyperparameters affect outcomes. (Section 4).
- 66 Despite some limitations of our approach discussed in Section 5, we conclude in Section 6 that counterfactual training
 67 provides a practical framework for researchers and practitioners interested in making opaque models more trustworthy.
 68 We also believe that this work serves as an opportunity for XAI researchers to re-evaluate the trend of improving XAI
 69 tools without improving the underlying models.

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

70 2 Related Literature

71 To the best of our knowledge, our proposed framework of counterfactual training represents the first attempt to use
 72 counterfactual explanations during training to improve model explainability. In high-level terms, we define model
 73 explainability as the extent to which valid explanations derived for an opaque model are also deemed plausible with
 74 respect to the underlying data and stakeholder requirements. To make this more concrete, we follow Augustin, Meinke,
 75 and Hein (2020) in tying the concept of explainability to the quality of counterfactual explanations that we can generate
 76 for a given model. The authors show that counterfactual explanations—understood here as minimal input perturbations
 77 that yield some desired model prediction—are generally more meaningful if the underlying model is more robust to
 78 adversarial examples. We can make intuitive sense of this finding when looking at adversarial training (AT) through
 79 the lens of representation learning with high degrees of freedom: by inducing models to “unlearn” representations that
 80 are susceptible to worst-case counterfactuals (i.e., adversarial examples), AT effectively removes some implausible
 81 explanations from the solution space.

82 2.1 Adversarial Examples are Counterfactual Explanations

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

93 In the natural language processing (NLP) domain, counterfactuals have similarly been used to improve models through
 94 data augmentation: Wu et al. (2021), propose *Polyjuice*, a general-purpose counterfactual generator for language mod-
 95 els. They demonstrate empirically that augmenting training data through *Polyjuice* counterfactuals improves robust-
 96 ness in a number of NLP tasks. Balashankar et al. (2023) also use *Polyjuice* to augment NLP datasets through diverse
 97 counterfactuals and show that classifier robustness improves up to 20%. Finally, Luu and Inoue (2023) introduce
 98 Counterfactual Adversarial Training (CAT), which also aims at improving generalization and robustness of language
 99 models. Specifically, they propose to proceed as follows: firstly, they identify training samples that are subject to
 100 high predictive uncertainty; secondly, they generate counterfactual explanations for those samples; and, finally, they
 101 fine-tune the given language model on the augmented dataset that includes the generated counterfactuals.

102 There have also been several attempts at formalizing the relationship between counterfactual explanations and adver-
 103 sarial examples (AE). Pointing to clear similarities in how CE and AE are generated, Freiesleben (2022) makes the
 104 case for jointly studying the opaqueness and robustness problem in representation learning. Formally, AE can be seen
 105 as the subset of CE for which misclassification is achieved (Freiesleben 2022). Similarly, Pawelczyk et al. (2022)
 106 show that CE and AE are equivalent under certain conditions and derive theoretical upper bounds on the distances
 107 between them.

108 Two recent works are closely related to ours in that they use counterfactuals during training with the explicit goal
 109 of affecting certain properties of post-hoc counterfactual explanations. Firstly, Ross, Lakkaraju, and Bastani (2024)
 110 propose a way to train models that are guaranteed to provide recourse for individuals to move from an adverse outcome
 111 to some positive target class with high probability. Their approach builds on adversarial training, where in this context
 112 susceptibility to targeted adversarial examples for the positive class is explicitly induced. The proposed method allows
 113 for imposing a set of actionability constraints ex-ante: for example, users can specify that certain features (e.g., *age*,
 114 *gender*, ...) are immutable. Secondly, Guo, Nguyen, and Yadav (2023) are the first to propose an end-to-end training
 115 pipeline that includes counterfactual explanations as part of the training procedure. In particular, they propose a
 116 specific network architecture that includes a predictor and CE generator network, where the parameters of the CE
 117 generator network are learnable. Counterfactuals are generated during each training iteration and fed back to the
 118 predictor network. In contrast to Guo, Nguyen, and Yadav (2023), we impose no restrictions on the neural network
 119 architecture at all.

120 2.2 Beyond Robustness

121 Improving the adversarial robustness of models is not the only path towards aligning representations with plausible
 122 explanations. In a work closely related to this one, Altmeyer et al. (2024) show that explainability can be improved
 123 through model averaging and refined model objectives. The authors propose a way to generate counterfactuals that
 124 are maximally **faithful** to the model in that they are consistent with what the model has learned about the underlying

125 data. Formally, they rely on tools from energy-based modelling to minimize the divergence between the distribution
 126 of counterfactuals and the conditional posterior over inputs learned by the model. Their proposed counterfactual
 127 explainer, *ECCo*, yields plausible explanations if and only if the underlying model has learned representations that
 128 align with them. They find that both deep ensembles ([Lakshminarayanan, Pritzel, and Blundell 2017](#)) and joint energy-
 129 based models (JEMs) ([Grathwohl et al. 2020](#)) tend to do well in this regard.

130 Once again it helps to look at these findings through the lens of representation learning with high degrees of freedom.
 131 Deep ensembles are approximate Bayesian model averages, which are most called for when models are underspecified
 132 by the available data ([Wilson 2020](#)). Averaging across solutions mitigates the aforementioned risk of relying on a
 133 single locally optimal representations that corresponds to semantically meaningless explanations for the data. Previous
 134 work by Schut et al. ([2021](#)) similarly found that generating plausible (“interpretable”) counterfactual explanations is
 135 almost trivial for deep ensembles that have also undergone adversarial training. The case for JEMs is even clearer:
 136 they involve a hybrid objective that induces both high predictive performance and generative capacity ([Grathwohl et al.
 137 2020](#)). This is closely related to the idea of aligning models with plausible explanations and has inspired our proposed
 138 counterfactual training objective, as we explain in Section 3.

139 3 Counterfactual Training

140 Counterfactual training combines ideas from adversarial training, energy-based modelling and counterfactuals expla-
 141 nations with the explicit objective of aligning representations with plausible explanations that comply with user re-
 142 quirements. In the context of CEs, plausibility has broadly been defined as the degree to which counterfactuals comply
 143 with the underlying data generating process ([Poyiadzi et al. 2020; Guidotti 2022; Altmeyer et al. 2024](#)). Plausibility
 144 is a necessary but insufficient condition for using CEs to provide algorithmic recourse (AR) to individuals affected
 145 by opaque models in practice. This is because for recourse recommendations to be **actionable**, they need to not only
 146 result in plausible counterfactuals but also be attainable. A plausible CE for a rejected 20-year-old loan applicant, for
 147 example, might reveal that their application would have been accepted, if only they were 20 years older. Ignoring
 148 all other features, this would comply with the definition of plausibility if 40-year-old individuals were in fact more
 149 credit-worthy on average than young adults. But of course this CE does not qualify for providing actionable recourse
 150 to the applicant since *age* is not a (directly) mutable feature. For our intents and purposes, counterfactual training aims
 151 to improve model explainability by aligning models with counterfactuals that meet both desiderata, plausibility and
 152 actionability. Formally, we define explainability as follows:

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

157 We say that M_θ is **explainable** to the extent that faithfully generated counterfactuals are plausible (i.e. consistent with
 158 the data) and actionable. Formally, we define these properties as follows:

- 159 1. (Plausibility) $\int^A p(\mathbf{x}' | \mathbf{y}^+) d\mathbf{x} \rightarrow 1$ where A is some small region around \mathbf{x}' .
 160 2. (Actionability) Permutations Δ are subject to some actionability constraints.

161 We consider counterfactuals as faithful to the extent that they are consistent with what the model has learned about the
 162 input data. Let $p_\theta(\mathbf{x} | \mathbf{y}^+)$ denote the conditional posterior over inputs, then formally:

- 163 3. (Faithfulness) $\int^A p_\theta(\mathbf{x}' | \mathbf{y}^+) d\mathbf{x} \rightarrow 1$ where A is defined as above.

164 The definitions of faithfulness and plausibility in Definition 3.1 are the same as in Altmeyer et al. ([2024](#)), with adapted
 165 notation. Actionability constraints in Definition 3.1 vary and depend on the context in which M_θ is deployed. In this
 166 work, we focus on domain and mutability constraints for individual features x_d for $d = 1, \dots, D$. We limit ourselves
 167 to classification tasks for reasons discussed in Section 5.

168 3.1 Our Proposed Objective

169 Let \mathbf{x}'_t for $t = 0, \dots, T$ denote a counterfactual explanation generated through gradient descent over T iterations
 170 as initially proposed by Wachter, Mittelstadt, and Russell ([2017](#)). For our purposes, we let T vary and consider the
 171 counterfactual search as converged as soon as the predicted probability for the target class has reached a pre-determined
 172 threshold, $\tau : \mathcal{S}(M_\theta(\mathbf{x}'))[y^+] \geq \tau$, where \mathcal{S} is the softmax function.²

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

173 To train models with high explainability as defined in Definition 3.1, we propose to leverage counterfactuals in the
 174 following objective:

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

175 where $\text{yloss}(\cdot)$ is any conventional classification loss that induces discriminative performance (e.g., cross-entropy).
 176 The second and third terms in Equation 1 are explained in more detail below. For now, they can be sufficiently de-
 177 scribed as inducing explainability directly and indirectly by penalizing: (1) the contrastive divergence, $\text{div}(\cdot)$, between
 178 mature counterfactuals \mathbf{x}'_T and observed samples x and, (2) the adversarial loss, $\text{advloss}(\cdot)$, with respect to nascent
 179 counterfactuals $\mathbf{x}'_{t \leq T}$. Finally, $\text{ridge}(\cdot)$ denotes a Ridge penalty (ℓ_2 -norm) that regularises the magnitude of the energy
 180 terms involved in $\text{div}(\cdot)$ (Du and Mordatch 2020). The tradeoff between the different components can be governed by
 181 adjusting the strengths of the penalties λ_{div} , λ_{adv} and λ_{reg} .

182 3.1.1 Directly Inducing Explainability through Contrastive Divergence

183 Grathwohl et al. (2020) observe that any classifier can be re-interpreted as a joint energy-based model (JEM) that
 184 learns to discriminate output classes conditional on the observed (training) samples from $p(\mathbf{x})$ and the generated
 185 samples from $p_{\theta}(\mathbf{x})$. They show that JEMs can be trained to perform well at both tasks by directly maximizing the
 186 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 factor can be optimized using
 187 conventional cross-entropy as in Equation 1. Then, to optimize $\log p_{\theta}(\mathbf{x})$ Grathwohl et al. (2020) minimize the
 188 contrastive divergence between these observed samples from $p(\mathbf{x})$ and generated samples from $p_{\theta}(\mathbf{x})$.

189 A key empirical finding in Altmeyer et al. (2024) was that JEMs tend to do well with respect to the plausibility ob-
 190 jective in Definition 3.1. If we consider samples drawn from $p_{\theta}(\mathbf{x})$ as counterfactuals, this is an expected finding,
 191 because the JEM objective effectively minimizes the divergence between the conditional posterior and $p(\mathbf{x}|y^+)$. To
 192 generate samples, Grathwohl et al. (2020) rely on Stochastic Gradient Langevin Dynamics (SGLD) using an uninfor-
 193 mative prior for initialization. This is where we depart from their methodology: instead of SGLD, we propose to use
 194 counterfactual explainers to generate counterfactuals of observed training samples. Specifically, we have:

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

195 where $\mathcal{E}_{\theta}(\cdot)$ denotes the energy function. In particular, we set $\mathcal{E}_{\theta}(\mathbf{x}, y) = -\mathbf{M}_{\theta}(\mathbf{x}^+)[y^+]$ where y^+ denotes the index
 196 of the randomly drawn target class, $y^+ \sim p(y)$, and \mathbf{x}^+ denotes an observed data point sampled from target domain:
 197 $\mathbf{X}^+ = \{\mathbf{x} : y = y^+\}$. Conditional on the target class y^+ , \mathbf{x}'_T denotes a mature counterfactual for a randomly sampled
 198 factual from a non-target class generated through a gradient-based counterfactual generator for at most T iterations.
 199 We define mature counterfactuals as those that have either exhausted T or reached convergence in terms of the pre-
 200 determined decision threshold earlier.

201 Intuitively, the gradient of Equation 2 decreases the energy of observed training samples (positive samples) while at
 202 same time increasing the energy of counterfactuals (negative samples) (Du and Mordatch 2020). As the generated
 203 counterfactuals get more plausible (Definition 3.1) over the course of training, these two opposing effects gradually
 204 balance each out (Lippe 2024).

205 The departure from SGLD allows us to tap into the vast repertoire of explainers that have been proposed in the literature
 206 to meet different desiderata. Typically, these methods facilitate the imposition of domain and mutability constraints,
 207 for example. In principle, any existing approach for generating counterfactual explanations is viable, so long as it does
 208 not violate the faithfulness condition. Like JEMs (Murphy 2022), counterfactual training can be considered as a form
 209 of contrastive representation learning.

210 3.1.2 Indirectly Inducing Explainability through Adversarial Robustness

211 Based on our analysis in Section 2, counterfactuals \mathbf{x}' can be repurposed as additional training samples (Luu and Inoue
 212 2023; Balashankar et al. 2023) or adversarial examples (Freiesleben 2022; Pawelczyk et al. 2022). This leaves some
 213 flexibility with respect to the exact choice for $\text{advloss}(\cdot)$ in Equation 1. An intuitive functional form to use, though
 214 likely not the only reasonable choice, is inspired by adversarial training:

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

215 Under this choice, we consider nascent counterfactuals $\mathbf{x}'_{t \leq T}$ as adversarial examples as long as the magnitude of the
 216 perturbation to any individual feature is at most ε . This is closely aligned with Szegedy et al. (2013), who define an

adversarial attack as an “imperceptible non-random perturbation”. Thus, we choose to work with a different distinction between CE and AE than Freiesleben (2022), who considers misclassification as the key distinguishing feature of AE. One of the key observations in this work is that we can leverage counterfactual explanations during training and get adversarial examples, essentially for free.

3.2 Encoding Actionability Constraints

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

Since actionability constraints are binding at test time, we should also impose them when generating \mathbf{x}' during each training iteration to align model representations with user requirements. Through their effect on \mathbf{x}' , both types of constraints influence model outcomes through Equation 2. Here it is crucial that we avoid penalizing implausibility that arises due to mutability constraints. For any mutability-constrained feature d this can be achieved by enforcing $\mathbf{x}[d] - \mathbf{x}'[d] := 0$ whenever perturbing $\mathbf{x}'[d]$ in the direction of $\mathbf{x}[d]$ would violate mutability constraints. Specifically, we set $\mathbf{x}[d] := \mathbf{x}'[d]$ if:

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

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

Proposition 3.1 (Protecting Immutable Features). *Let $f_\theta(\mathbf{x}) = \mathcal{S}(\mathbf{M}_\theta(\mathbf{x})) = \mathcal{S}(\Theta\mathbf{x})$ denote a linear classifier with softmax activation \mathcal{S} (i.e., multinomial logistic regression) where $y \in \{1, \dots, K\} = \mathcal{K}$ and $\mathbf{x} \in \mathbb{R}^D$. If we assume multivariate Gaussian class densities with common diagonal covariance matrix $\Sigma_k = \Sigma$ for all $k \in \mathcal{K}$, then protecting an immutable feature from the contrastive divergence penalty (Equation 2) will result in lower classifier sensitivity to that feature relative to the remaining features, provided that at least one of those is discriminative and mutable.*

It is worth highlighting that Proposition 3.1 assumes independence of features. This raises a valid concern about the effect of protecting immutable features in the presence of proxy features that remain unprotected. We discuss this limitation in Section 5.

3.3 Illustration

To better convey the intuition underlying our proposed method, we illustrate different model outcomes in Example 3.1.

Example 3.1 (Prediction of Consumer Credit Default). Suppose we are interested in predicting the likelihood that loan applicants default on their credit. We have access to historical data on previous loan takers comprised of a binary outcome variable ($y \in \{1 = \text{default}, 2 = \text{no default}\}$) two input features: (1) the subjects’ age, which we define as immutable, and (2) the subjects’ existing level of debt, which we define as mutable.

We have simulated this scenario using synthetic data with independent features and Gaussian class-conditional densities in Figure 1. The four panels in Figure 1 show the outcomes for different training procedures using the same model architecture each time (a linear classifier). In each case, we show the linear decision boundary (green) and the training data colored according to their ground-truth label: orange points belong to the target class, $y^+ = 2$, blue points belong to the non-target class, $y^- = 1$. Stars indicate counterfactuals in the target class generated at test time using generic gradient descent until convergence.

In panel (a), we have trained our model conventionally, and we do not impose mutability constraints at test time. The generated counterfactuals are all valid, but not plausible: they are clearly distinguishable from the ground-truth data. In panel (b), we have trained our model with counterfactual training, once again not imposing mutability constraints at test time. We observe that the counterfactuals are clearly plausible, therefore meeting the first objective of Definition 3.1.

³For the proof, see the supplementary appendix.

266 In panel (c), we have used conventional training again, this time imposing the mutability constraint on *age* at test time.
 267 Counterfactuals are valid but involve some substantial reductions in *debt* for some individuals (very young applicants).
 268 By comparison, counterfactual paths are shorter on average in panel (d), where we have used counterfactual training
 269 and protected immutable features as described in Section 3.2. In particular, we observe that due to the classifier's
 270 lower sensitivity to *age*, recourse recommendations with respect to *debt* are much more homogenous, in that they do
 271 not disproportionately punish younger individuals. The counterfactuals are also plausible with respect to the mutable
 272 feature. Thus, we consider the model in panel (d) as the most explainable according to Definition 3.1.

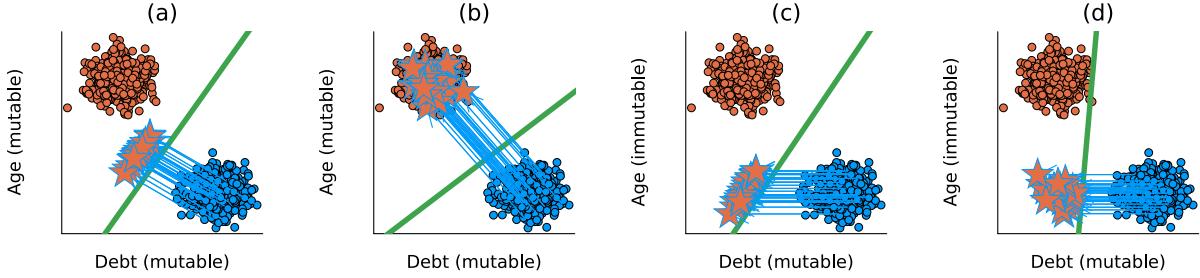


Figure 1: Visual illustration of how counterfactual training improves explainability. See Example 3.1 for details.

273 4 Experiments

274 In this section, we present experiments that we have conducted in order to answer the following research questions:

275 **Research Question 4.1** (Plausibility). *Does our proposed counterfactual training objective (Equation 1) induce mod-
 276 els to learn plausible explanations?*

277 **Research Question 4.2** (Actionability). *Does our proposed counterfactual training objective (Equation 1) yield more
 278 favorable algorithmic recourse outcomes in the presence of actionability constraints?*

279 Beyond this, we are also interested in understanding how robust our answers to RQ 4.1 and RQ 4.2 are:

280 **Research Question 4.3** (Hyperparameters). *What are the effects of different hyperparameter choices with respect to
 281 Equation 1?*

282 4.1 Experimental Setup

283 4.1.1 Evaluation

284 Our key outcome of interest is how well models perform with respect to explainability (Definition 3.1): to this end, we
 285 focus primarily on the plausibility and cost of faithfully generated counterfactuals at test time. To measure the cost of
 286 counterfactuals, we follow the standard convention of using distances (ℓ_1 -norm) between factuals and counterfactuals
 287 as a proxy. For plausibility, we assess how similar counterfactuals are to observed samples in the target domain. We
 288 rely on the distance-based metric used by Altmeyer et al. (2024),

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

289 and introduce a novel divergence metric,

$$\text{implaus}_{\text{div}}(\mathbf{X}', \mathbf{X}^+) = \text{MMD}(\mathbf{X}', \mathbf{X}^+) \quad (5)$$

290 where \mathbf{X}' denotes a set of multiple counterfactuals and $\text{MMD}(\cdot)$ is an unbiased estimate of the squared population
 291 maximum mean discrepancy (Gretton et al. 2012). The metric in Equation 5 is equal to zero iff $\mathbf{X}' = \mathbf{X}^+$.

292 In addition to cost and plausibility, we also compute other standard metrics to evaluate counterfactuals at test time in-
 293 cluding validity and redundancy. Finally, we also assess the predictive performance of models using standard metrics.

294 We run the experiments with three CE generators: *Generic* of Wachter, Mittelstadt, and Russell (2017) as a simple
 295 baseline approach, *REVISE* (Joshi et al. 2019) that aims to generate plausible counterfactuals using a surrogate Vari-
 296 ational Autoencoder (VAE), and *ECCo*—the generator of Altmeyer et al. (2023) but without the conformal prediction
 297 component—as a method that directly targets both faithfulness and plausibility of the CEs.

298 **4.2 Experimental Results**299 **4.2.1 Plausibility**300 **4.2.2 Actionability**301 **4.2.3 Impact of hyperparameter settings**

302 We extensively test the impact of three types of hyperparameters on the proposed training regime. Our complete results
 303 are available in the technical appendix; this section focuses on the main findings.

304 **Hyperparameters of the CE generators.** First, we observe that CT is highly sensitive to hyperparameter settings but
 305 (a) there are manageable patterns and (b) we can typically identify settings that improve either plausibility or cost, and
 306 commonly both of them at the same time. Second, we note that the choice of a CE generator has a major impact on
 307 the results. For example, *REVISE* tends to perform the worst, most likely because it uses a surrogate VAE to generate
 308 counterfactuals which impedes faithfulness (Altmeyer et al. 2024). Third, increasing T , the maximum number of
 309 steps, generally yields better outcomes because more CEs can mature in each training epoch. Fourth, the impact of τ ,
 310 the required decision threshold is more difficult to predict. On “harder” datasets it may be difficult to satisfy high τ for
 311 any given sample (i.e., also factuals) and so increasing this threshold does not seem to correlate with better outcomes.
 312 In fact, we have generally found that a choice of $\tau = 0.5$ leads to optimal results because it is associated with high
 313 proportions of mature counterfactuals.

314 **Hyperparameters for penalties.** We find that the strength of the energy regularization, λ_{reg} is highly impactful; energy
 315 must be sufficiently regularized to avoid poor performance in terms of decreased plausibility and increased costs. The
 316 sensitivity with respect to λ_{div} and λ_{adv} is much less evident. While high values of λ_{reg} may increase the variability
 317 in outcomes when combined with high values for any of the other penalties in Equation 1, this effect is not very
 318 pronounced.

319 **Other hyperparameters.** We observe that the effectiveness and stability of CT is positively associated with the number
 320 of counterfactuals generated during each training epoch. We also confirm that a higher number of training epochs is
 321 beneficial. Interestingly, we find that it is not necessary to employ CT during the entire training phase to achieve the
 322 desired improvements in explainability. We have tested training models conventionally during the first half of training
 323 before switching to CT after this initial “burn-in” period and observed positive results. Put differently, CT may be a
 324 way to improve the explainability of trained models in a fine-tuning manner.

325 **5 Discussion**

326 We begin the discussion by addressing the direct extensions of the counterfactual training approach in Section 5.1.
 327 Then, we look at its broader limitations and challenges in Section 5.2.

328 **5.1 Future research**

329 **CT is defined only for classification settings.** Our formulation relies on the distinction between non-target class(es)
 330 y^- and target class(es) y^+ to generate counterfactuals through Equation 1. While y^- and y^+ can be arbitrarily defined
 331 by the user, CT requires the output space \mathcal{Y} to be discrete. Thus, it applies to binary and multi-class classification but
 332 it is not well-defined for other ML tasks where the change in outcome with respect to a decision threshold τ cannot
 333 be readily quantified. In fact, this is a common restriction in research on CEs and AR that predominantly focuses on
 334 classification models. Although other settings have attracted some interest (e.g., regression in Spooner et al. 2021;
 335 Zhao, Broelemann, and Kasneci 2023), there is still no consensus on what constitutes a counterfactual in such settings.

336 **CT is subject to training instabilities.** Joint energy-based models are susceptible to instabilities during training (Grath-
 337 wohl et al. 2020) and even though we depart from the SGLD-based sampling, we still encounter major variability in
 338 the outcomes. CT is exposed to two potential sources of instabilities: (1) the energy-based contrastive divergence term
 339 in Equation 2, and (2) the underlying counterfactual explainers. For example, Altmeyer et al. (2023) recognize this
 340 to be a challenge for *ECCCo* and so it may have downstream impacts on our proposed method. Still, we find that
 341 training instabilities can be successfully mitigated by regularizing energy (λ_{reg}), generating a sufficiently large number
 342 of counterfactuals during each training epoch and including only mature counterfactuals for contrastive divergence.

343 **CT is sensitive to hyperparameter selection.** As discussed in Section 4.2.3, our method benefits from tuning certain
 344 key hyperparameters. In this work, we have relied exclusively on grid search for this task. Future work on CT could
 345 benefit from investigating more sophisticated approaches towards hyperparameter tuning. Notably, counterfactual
 346 training is iterative which makes a variety of methods applicable, including Bayesian (e.g., Snoek, Larochelle, and
 347 Adams 2012) or gradient-based (e.g., Franceschi et al. 2017) optimization.

348 **5.2 Limitations and challenges**

349 ***CT increases the training time of models.*** Counterfactual training promotes explainability through CEs and robustness
 350 through AEs at the cost of longer training times compared to conventional training regimes. While higher numbers
 351 of iterations and counterfactuals per iteration positively impact the quality of found solutions, they also increase the
 352 required amount of computations. We find that relatively small grids with 270 settings can take almost four hours for
 353 more demanding datasets on a high-performance computing cluster with 34 2GB CPUs⁴. However, there are three
 354 factors that attenuate the impact of this limitation. First, CT provides counterfactual explanations for the training
 355 samples essentially for free, which may be beneficial in many ADM systems. Second, we find that CT can retain its
 356 value when used as a “fine-tuning” training regime for conventionally-trained models. Third, in principle, CT yields
 357 itself to parallel execution, which we have leveraged for our own experiments.

358 ***Immutable features may have proxies.*** In Proposition 3.1 we define an approach to protect immutable features and
 359 thus increase the actionability of the generated counterfactuals. However, this approach requires that model owners
 360 define the mutability constraints for (all) features considered by the model. Even with sufficient domain knowledge
 361 to protect all immutable features—ones that cannot be changed at all and ones that cannot be reasonably expected
 362 to change—there may exist proxies that are theoretically mutable (and hence should not be protected) but preserve
 363 enough information about the principals to counteract the protections. As one example, consider the Adult dataset
 364 used in our experiments where the mutable education status is a proxy for the immutable age, in that the attainment of
 365 degrees is correlated with age. Delineating actionability is a major undecided challenge in the AR literature (see, e.g.,
 366 Venkatasubramanian and Alfano 2020) impacting the capacity of CT to increase the explainability of the model.

367 ***Interventions on features may have downstream impacts on fairness.***

- 368 - Equality of opportunity?
- 369 - Social segregation?
- 370 - Supporting AR within one context/system may still unfairly target recourse-affected individuals within other contexts/systems? In Example 3.1 younger individuals gain access to a loan but their age could still lead to, e.g., enforcing
 371 stronger supervisory mechanisms?

373 **6 Conclusion**

374 **References**

- 375 Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. “Counterfactual
 376 Vision and Language Learning.” In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition
 377 (CVPR)*, 10041–51. <https://doi.org/10.1109/CVPR42600.2020.01006>.
- 378 Altmeyer, Patrick, Arie van Deursen, et al. 2023. “Explaining Black-Box Models Through Counterfactuals.” In
 379 *Proceedings of the JuliaCon Conferences*, 1:130. 1.
- 380 Altmeyer, Patrick, Mojtaba Farmanbar, Arie van Deursen, and Cynthia C. S. Liem. 2023. “Faithful Model Explanations
 381 Through Energy-Constrained Conformal Counterfactuals.” <https://arxiv.org/abs/2312.10648>.
- 382 Altmeyer, Patrick, Mojtaba Farmanbar, Arie van Deursen, and Cynthia CS Liem. 2024. “Faithful Model Explanations
 383 Through Energy-Constrained Conformal Counterfactuals.” In *Proceedings of the AAAI Conference on Artificial
 384 Intelligence*, 38:10829–37. 10.
- 385 Augustin, Maximilian, Alexander Meinke, and Matthias Hein. 2020. “Adversarial Robustness on in-and Out-
 386 Distribution Improves Explainability.” In *European Conference on Computer Vision*, 228–45. Springer.
- 387 Balashankar, Ananth, Xuezhi Wang, Yao Qin, Ben Packer, Nithum Thain, Ed Chi, Jilin Chen, and Alex Beutel. 2023.
 388 “Improving Classifier Robustness Through Active Generative Counterfactual Data Augmentation.” In *Findings of
 389 the Association for Computational Linguistics: EMNLP 2023*, 127–39.
- 390 Bezanson, Jeff, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. “Julia: A Fresh Approach to Numerical
 391 Computing.” *SIAM Review* 59 (1): 65–98. <https://doi.org/10.1137/141000671>.
- 392 Bouchet-Valat, Milan, and Bogumił Kamiński. 2023. “DataFrames.jl: Flexible and Fast Tabular Data in Julia.” *Journal
 393 of Statistical Software* 107 (4): 1–32. <https://doi.org/10.18637/jss.v107.i04>.
- 394 Byrne, Simon, Lucas C. Wilcox, and Valentin Churavy. 2021. “MPI.jl: Julia Bindings for the Message Passing
 395 Interface.” *Proceedings of the JuliaCon Conferences* 1 (1): 68. <https://doi.org/10.21105/jcon.00068>.
- 396 Chagas, Ronan Arraes Jardim, Ben Baumgold, Glen Hertz, Hendrik Ranocha, Mark Wells, Nathan Boyer, Nicholas
 397 Ritchie, et al. 2024. “Ronisbr/PrettyTables.jl: V2.4.0.” Zenodo. <https://doi.org/10.5281/zenodo.13835553>.
- 398 Christ, Simon, Daniel Schwabedener, Christopher Rackauckas, Michael Krabbe Borregaard, and Thomas Breloff.
 399 2023. “Plots.jl – a User Extendable Plotting API for the Julia Programming Language.” <https://doi.org/https://doi.org/10.5334/jors.431>.

⁴See supplementary appendix for computational details.

- 401 Danisch, Simon, and Julius Krumbiegel. 2021. “Makie.jl: Flexible High-Performance Data Visualization for Julia.”
 402 *Journal of Open Source Software* 6 (65): 3349. <https://doi.org/10.21105/joss.03349>.
- 403 Du, Yilun, and Igor Mordatch. 2020. “Implicit Generation and Generalization in Energy-Based Models.” <https://arxiv.org/abs/1903.08689>.
- 404 Franceschi, Luca, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. 2017. “Forward and Reverse Gradient-
 405 Based Hyperparameter Optimization.” In *Proceedings of the 34th International Conference on Machine Learning*,
 406 edited by Doina Precup and Yee Whye Teh, 70:1165–73. Proceedings of Machine Learning Research. PMLR.
 407 <https://proceedings.mlr.press/v70/franceschi17a.html>.
- 408 Frankle, Jonathan, and Michael Carbin. 2019. “The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural
 409 Networks.” In *International Conference on Learning Representations*. <https://openreview.net/forum?id=rJLb3RcF7>.
- 410 Freiesleben, Timo. 2022. “The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples.”
 411 *Minds and Machines* 32 (1): 77–109.
- 412 Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy. 2014. “Explaining and Harnessing Adversarial Examples.”
 413 <https://arxiv.org/abs/1412.6572>.
- 414 Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- 415 Grathwohl, Will, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swer-
 416 sky. 2020. “Your Classifier Is Secretly an Energy Based Model and You Should Treat It Like One.” In *International
 417 Conference on Learning Representations*.
- 418 Gretton, Arthur, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. “A Kernel
 419 Two-Sample Test.” *The Journal of Machine Learning Research* 13 (1): 723–73.
- 420 Guidotti, Riccardo. 2022. “Counterfactual Explanations and How to Find Them: Literature Review and Benchmark-
 421 ing.” *Data Mining and Knowledge Discovery*, 1–55.
- 422 Guo, Hangzhi, Thanh H. Nguyen, and Amulya Yadav. 2023. “CounterNet: End-to-End Training of Prediction Aware
 423 Counterfactual Explanations.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery
 424 and Data Mining*, 577–89. KDD ’23. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3580305.3599290>.
- 425 Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer New
 426 York. <https://doi.org/10.1007/978-0-387-84858-7>.
- 427 Innes, Michael, Elliot Saba, Keno Fischer, Dhairyा Gandhi, Marco Conchetto Rudilosso, Neethu Mariya Joy, Tejan
 428 Karmali, Avik Pal, and Viral Shah. 2018. “Fashionable Modelling with Flux.” <https://arxiv.org/abs/1811.01457>.
- 429 Innes, Mike. 2018. “Flux: Elegant Machine Learning with Julia.” *Journal of Open Source Software* 3 (25): 602.
 430 <https://doi.org/10.21105/joss.00602>.
- 431 Joshi, Shalmali, Oluwasanmi Koyejo, Warut Vigitbenjaronk, Been Kim, and Joydeep Ghosh. 2019. “Towards Realistic
 432 Individual Recourse and Actionable Explanations in Black-Box Decision Making Systems.” <https://arxiv.org/abs/1907.09615>.
- 433 Kolter, Zico. 2023. “Keynote Addresses: SaTML 2023 .” In *2023 IEEE Conference on Secure and Trustworthy
 434 Machine Learning (SaTML)*, xvi–. Los Alamitos, CA, USA: IEEE Computer Society. <https://doi.org/10.1109/SaTML54575.2023.00009>.
- 435 Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. “Simple and Scalable Predictive Uncer-
 436 tainty Estimation Using Deep Ensembles.” *Advances in Neural Information Processing Systems* 30.
- 437 Lippe, Phillip. 2024. “UvA Deep Learning Tutorials.” <https://uvadlc-notebooks.readthedocs.io/en/latest/>.
- 438 Luu, Hoai Linh, and Naoya Inoue. 2023. “Counterfactual Adversarial Training for Improving Robustness of Pre-
 439 Trained Language Models.” In *Proceedings of the 37th Pacific Asia Conference on Language, Information and
 440 Computation*, 881–88.
- 441 McGregor, Sean. 2021. “Preventing repeated real world AI failures by cataloging incidents: The AI incident database.”
 442 In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:15458–63. 17.
- 443 Morcos, Ari S., Haonan Yu, Michela Paganini, and Yuandong Tian. 2019. “One Ticket to Win Them All: Gener-
 444 alizing Lottery Ticket Initializations Across Datasets and Optimizers.” In *Proceedings of the 33rd International
 445 Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.
- 446 Murphy, Kevin P. 2022. *Probabilistic Machine Learning: An Introduction*. MIT Press.
- 447 O’Neil, Cathy. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*.
 448 Crown.
- 449 Pawelczyk, Martin, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. “Exploring
 450 Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis.”
 451 In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, edited by Gustau
 452 Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, 151:4574–94. Proceedings of Machine Learning Research.
 453 PMLR. <https://proceedings.mlr.press/v151/pawelczyk22a.html>.

- 459 Poyiadzi, Rafael, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. “FACE: Feasible and
 460 Actionable Counterfactual Explanations.” In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*,
 461 344–50.
- 462 Ross, Alexis, Himabindu Lakkaraju, and Osbert Bastani. 2024. “Learning Models for Actionable Recourse.” In
 463 *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS ’21. Red
 464 Hook, NY, USA: Curran Associates Inc.
- 465 Sauer, Axel, and Andreas Geiger. 2021. “Counterfactual Generative Networks.” <https://arxiv.org/abs/2101.06046>.
- 466 Schut, Lisa, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. 2021. “Generating
 467 Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties.” In
 468 *International Conference on Artificial Intelligence and Statistics*, 1756–64. PMLR.
- 469 Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. 2012. “Practical Bayesian Optimization of Machine Learning
 470 Algorithms.” In *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. J. Burges, L. Bottou,
 471 and K. Q. Weinberger. Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.
- 472 Spooner, Thomas, Danial Dervovic, Jason Long, Jon Shepard, Jiahao Chen, and Daniele Magazzeni. 2021. “Counter-
 473 factual Explanations for Arbitrary Regression Models.” *CoRR* abs/2106.15212. <https://arxiv.org/abs/2106.15212>.
- 474 Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.
 475 2013. “Intriguing Properties of Neural Networks.” <https://arxiv.org/abs/1312.6199>.
- 476 Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. “Learning What Makes a Difference from
 477 Counterfactual Examples and Gradient Supervision.” In *Computer Vision–ECCV 2020: 16th European Confer-
 478 ence, Glasgow, UK, August 23–28, 2020, Proceedings, Part x 16*, 580–99. Springer.
- 479 Venkatasubramanian, Suresh, and Mark Alfano. 2020. “The Philosophical Basis of Algorithmic Recourse.” In *Pro-
 480 ceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 284–93. FAT* ’20. New York,
 481 NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3351095.3372876>.
- 482 Wachter, Sandra, Brent Mittelstadt, and Chris Russell. 2017. “Counterfactual Explanations Without Opening the Black
 483 Box: Automated Decisions and the GDPR.” *Harv. JL & Tech.* 31: 841. <https://doi.org/10.2139/ssrn.3063289>.
- 484 Wilson, Andrew Gordon. 2020. “The Case for Bayesian Deep Learning.” <https://arxiv.org/abs/2001.10995>.
- 485 Wu, Tongshuang, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. “Polyjuice: Generating Counterfactuals
 486 for Explaining, Evaluating, and Improving Models.” In *Proceedings of the 59th Annual Meeting of the Associa-
 487 tion for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing
 488 (Volume 1: Long Papers)*, edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 6707–23. Online:
 489 Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.acl-long.523>.
- 490 Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. “Understanding Deep
 491 Learning (Still) Requires Rethinking Generalization.” *Commun. ACM* 64 (3): 107–15. <https://doi.org/10.1145/3446776>.
- 492 Zhao, Xuan, Klaus Broelemann, and Gjergji Kasneci. 2023. “Counterfactual Explanation for Regression via Disentan-
 493 glement in Latent Space.” In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, 976–84.
 494 Los Alamitos, CA, USA: IEEE Computer Society. <https://doi.org/10.1109/ICDMW60847.2023.00130>.

497 **G Notation**

- 498 • y^+ : The target class and also the index of the target class.
 499 • y^- : The non-target class and also the index of non-the target class.
 500 • \mathbf{y}^+ : The one-hot encoded output vector for the target class.
 501 • θ : Model parameters (unspecified).
 502 • Θ : Matrix of parameters.

503 **G.1 Other Technical Details**

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

504 **H Technical Details of Our Approach**

505 **H.1 Generating Counterfactuals through Gradient Descent**

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

508 **H.1.1 Background**

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

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

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

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

521 **H.1.2 Convergence**

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

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

531 **H.2 Protecting Mutability Constraints with Linear Classifiers**

532 In Section 3.2 we explain that to avoid penalizing implausibility that arises due to mutability constraints, we impose a
 533 point mass prior on $p(\mathbf{x})$ for the corresponding feature. We argue in Section 3.2 that this approach induces models to
 534 be less sensitive to immutable features and demonstrate this empirically in Section 4. Below we derive the analytical
 535 results in Proposition 3.1.

536 *Proof.* Let d_{mtbl} and d_{immtbl} denote some mutable and immutable feature, respectively. Suppose that $\mu_{y^-, d_{\text{immtbl}}} <$
 537 $\mu_{y^+, d_{\text{immtbl}}}$ and $\mu_{y^-, d_{\text{mtbl}}} > \mu_{y^+, d_{\text{mtbl}}}$, where $\mu_{k,d}$ denotes the conditional sample mean of feature d in class k . In words,
 538 we assume that the immutable feature tends to take lower values for samples in the non-target class y^- than in the
 539 target class y^+ . We assume the opposite to hold for the mutable feature.

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

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

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

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

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

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

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

545 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
 546 made above. Hence, we have that $(\theta_{y^-, d_{\text{immtbl}}} - \theta_{y^+, d_{\text{immtbl}}}) < 0$ and $(\theta_{y^-, d_{\text{mtbl}}} - \theta_{y^+, d_{\text{mtbl}}}) > 0$

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

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

550 and equal to zero everywhere else.

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

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

557 By the same rationale, the contrastive divergence penalty can generally be expected to increase $\theta_{y^-, d_{\text{mtbl}}}$ exacerbating
 558 $(\theta_{y^-, d_{\text{mtbl}}} - \theta_{y^+, d_{\text{mtbl}}}) > 0$. In words, this has the effect of making the classifier more sensitive to the mutable feature, in
 559 that it would be more likely to predict class y^- as opposed to y^+ for higher values of d_{mtbl} .

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

563 H.3 Domain Constraints

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

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

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

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

574 H.4 Training Details

575 In this section, we describe the training procedure in detail. While the details laid out here are not crucial for under-
 576 standing our proposed approach, they are of importance to anyone looking to implement counterfactual training.

577 I Details on Main Experiments

578 I.1 Final Hyperparameters

579 As discussed Section 4, CT is sensitive to certain hyperparameter choices. We study the effect of many hyperparame-
 580 ters extensively in Section J. For the main results, we tune a small set of key hyperparameters (Section K). The final
 581 choices for the main results are presented for each data set in Table A1 along with training, test and batch sizes.

Table A1: Final hyperparameters used for the main results for the different datasets.

Data	No. Train	No. Test	Batchsize	Decision Threshold	No. Counterfactuals	λ_{reg}
Adult	$2.6 \cdot 10^4$	$5.01 \cdot 10^3$	$1 \cdot 10^3$	0.75	$5 \cdot 10^3$	0.25
California Housing	$1.65 \cdot 10^4$	$3.1 \cdot 10^3$	$1 \cdot 10^3$	0.5	$5 \cdot 10^3$	0.25
Circles	$3.6 \cdot 10^3$	600	30	0.5	$1 \cdot 10^3$	0.5
Credit	$1.06 \cdot 10^4$	$1.92 \cdot 10^3$	$1 \cdot 10^3$	0.5	$5 \cdot 10^3$	0.25
GMSC	$1.34 \cdot 10^4$	$2.47 \cdot 10^3$	$1 \cdot 10^3$	0.5	$5 \cdot 10^3$	0.5
Linearly Separable	$3.6 \cdot 10^3$	600	30	0.5	$1 \cdot 10^3$	0.25
Moons	$3.6 \cdot 10^3$	600	30	0.9	$1 \cdot 10^3$	0.25
Overlapping	$3.6 \cdot 10^3$	600	30	0.5	$1 \cdot 10^3$	0.25

582 I.2 Qualitative Findings for Image Data

Note

Figure A2 shows much more plausible (faithful) counterfactuals for a model with CT than the model with conventional training (Figure A3). In fact, this is not even using *ECCo+* and still showing better results than the best results we achieved in our AAAI paper for JEM ensembles.

583

584 J Grid Searches

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

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

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

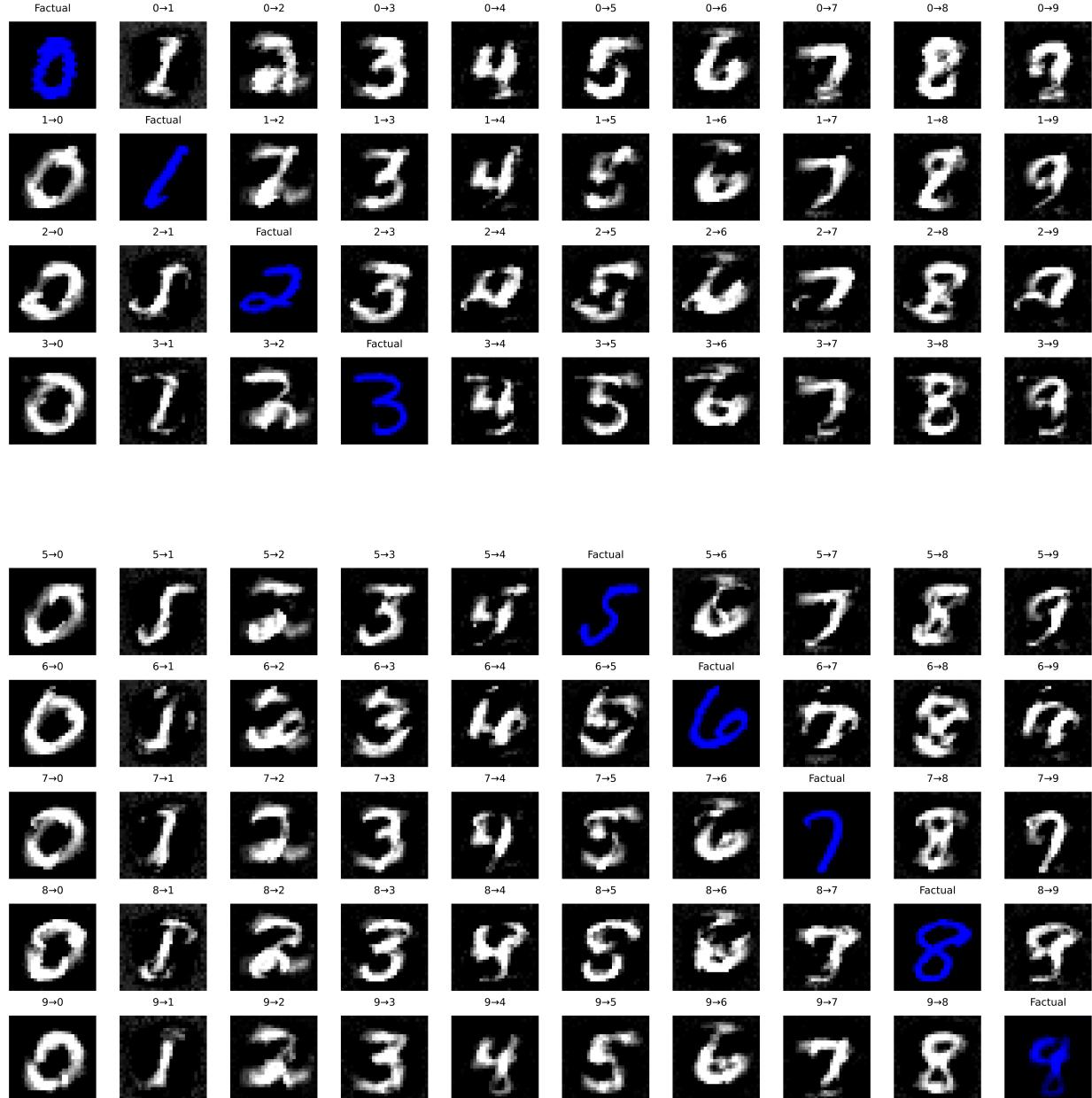


Figure A2: Counterfactual images for *MLP* with counterfactual training. The underlying generator, *ECCo*, aims to generate counterfactuals that are faithful to the model (Altmeyer et al. 2024).

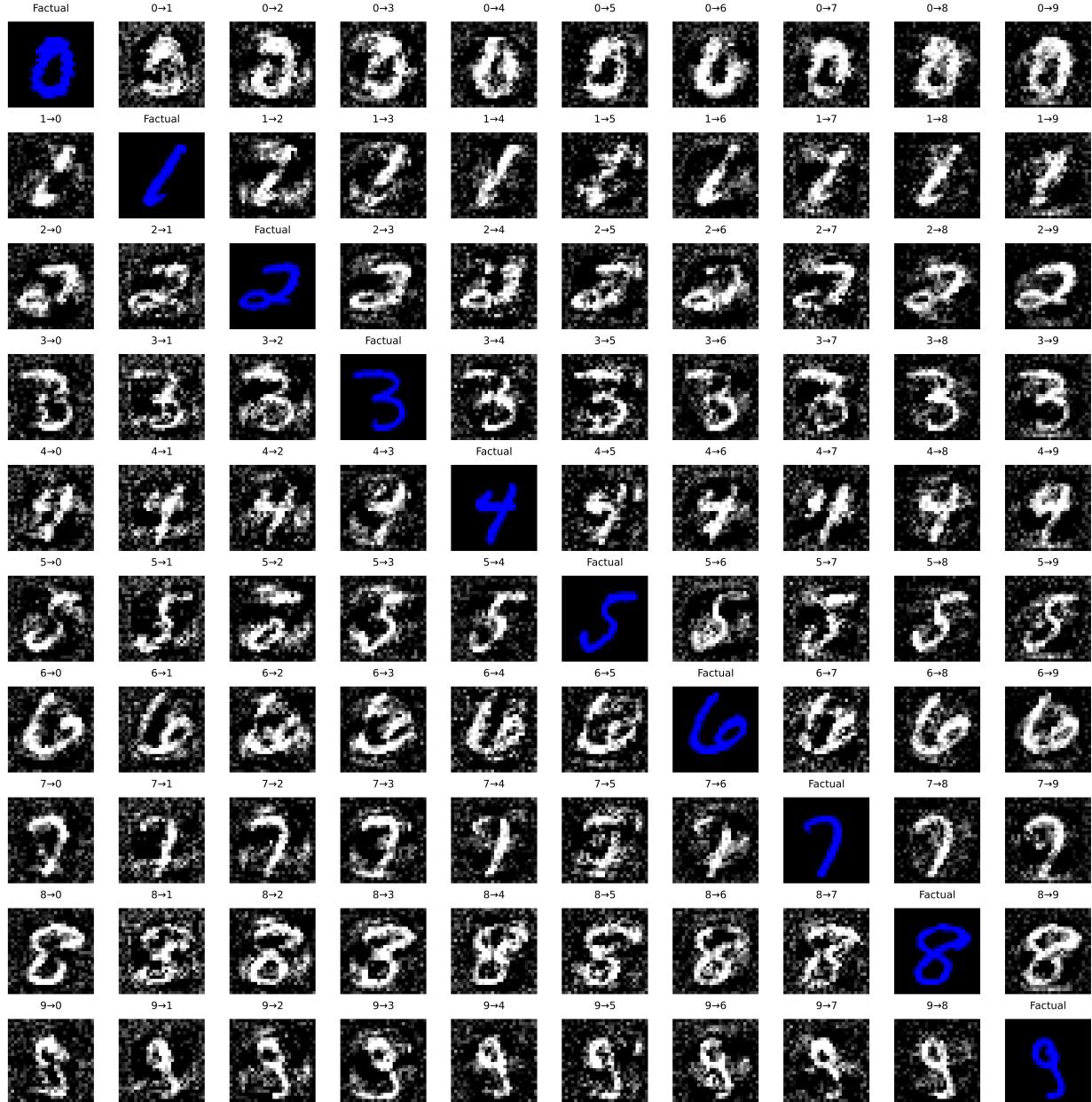


Figure A3: Counterfactual images for *MLP* with conventional training. The underlying generator, *ECCo*, aims to generate counterfactuals that are faithful to the model (Altmeyer et al. 2024).

596 **J.1 Evaluation Details**

597 To measure predictive performance, we compute the accuracy and F1-score for all models on test data (Table A2,
 598 Table A3, Table A4). With respect to explanatory performance, we report here our findings for the (im)plausibility
 599 and cost of counterfactuals at test time. Since the computation of our proposed divergence metric (Equation 5) is
 600 memory-intensive, we rely on the distance-based metric for the grid searches. For the counterfactual evaluation, we
 601 draw factual samples from the training data for the grid searches to avoid data leakage with respect to our final results
 602 reported in the body of the paper. Specifically, we want to avoid choosing our default hyperparameters based on results
 603 on the test data. Since we are optimizing for explainability, not predictive performance, we still present test accuracy
 604 and F1-scores.

605 **J.1.1 Predictive Performance**

606 We find that CT is associated with little to no decrease in average predictive performance for our synthetic datasets:
 607 test accuracy and F1-scores decrease by at most ~1 percentage point, but generally much less (Table A2, Table A3,
 608 Table A4). Variation across hyperparameters is negligible as indicated by small standard deviations for these metrics
 609 across the board.

610 **J.1.2 Counterfactual Outcomes**

611 Overall, we find that Counterfactual Training (CT) achieves its key objectives consistently across all hyperparameter
 612 settings and also broadly across datasets: plausibility is improved by up to ~60 percent (%) for the *Circles* data
 613 (e.g. Figure A4), ~25-30% for the *Moons* data (e.g. Figure A6) and ~10-20% for the *Linearly Separable* data (e.g.
 614 Figure A5). At the same time, the average costs of faithful counterfactuals are reduced in many cases by around
 615 ~20-25% for *Circles* (e.g. Figure A8) and up to ~50% for *Moons* (e.g. Figure A10). For the *Linearly Separable* data,
 616 costs are generally increased although typically by less than 10% (e.g. Figure A9), which reflects a common tradeoff
 617 between costs and plausibility (Altmeyer et al. 2024).

618 We do observe strong sensitivity to certain hyperparameters, with clear and manageable patterns. Concerning generator
 619 parameters, we firstly find that using *REVISE* to generate counterfactuals during training typically yields the worst
 620 outcomes out of all generators, often leading to a substantial decrease in plausibility. This finding can be attributed to
 621 the fact that *REVISE* effectively assigns the task of learning plausible explanations from the model itself to a surrogate
 622 VAE. In other words, counterfactuals generated by *REVISE* are less faithful to the model than *ECCo* and *Generic*, and
 623 hence we would expect them to be a less effective and, in fact, potentially detrimental role in our training regime.
 624 Secondly, we observe that allowing for a higher number of maximum steps T for the counterfactual search generally
 625 yields better outcomes. This is intuitive, because it allows more counterfactuals to reach maturity in any given iteration.
 626 Looking in particular at the results for *Linearly Separable*, it seems that higher values for T in combination with higher
 627 decision thresholds (τ) yields the best results when using *ECCo*. But depending on the degree of class separability
 628 of the underlying data, a high decision-threshold can also affect results adversely, as evident from the results for
 629 the *Overlapping* data (Figure A7): here we find that CT generally fails to achieve its objective because only a tiny
 630 proportion of counterfactuals ever reaches maturity.

631 Regarding penalty strengths, we find that the strength of the energy regularization, λ_{reg} is a key hyperparameter, while
 632 sensitivity with respect to λ_{div} and λ_{adv} is much less evident. In particular, we observe that not regularizing energy
 633 enough or at all typically leads to poor performance in terms of decreased plausibility and increased costs, in particular
 634 for *Circles* (Figure A12), *Linearly Separable* (Figure A13) and *Overlapping* (Figure A15). High values of λ_{reg} can
 635 increase the variability in outcomes, in particular when combined with high values for λ_{div} and λ_{adv} , but this effect is
 636 less pronounced.

637 Finally, concerning other hyperparameters we observe that the effectiveness and stability of CT is positively associated
 638 with the number of counterfactuals generated during each training epoch, in particular for *Circles* (Figure A20) and
 639 *Moons* (Figure A22). We further find that a higher number of training epochs is beneficial as expected, where we
 640 tested training models for 50 and 100 epochs. Interestingly, we find that it is not necessary to employ CT during
 641 the entire training phase to achieve the desired improvements in explainability: specifically, we have tested training
 642 models conventionally during the first half of training before switching to CT after this initial burn-in period.

643 **J.2 Generator Parameters**

644 The hyperparameter grid with varying generator parameters during training is shown in Note 1. The corresponding
 645 evaluation grid used for these experiments is shown in Note 2.

646 Note 1: Training Phase

- Generator Parameters:
 - Decision Threshold: 0.75, 0.9, 0.95
 - λ_{energy} : 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`

646

647 Note 2: Evaluation Phase

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

647

648 **J.2.1 Accuracy**

Table A2: Predictive performance measures by dataset and objective averaged across training-phase parameters (Note 1) and evaluation-phase parameters (Note 2).

Dataset	Variable	Objective	Mean	Std
Circles	Accuracy	Full	0.997	0.00309
Circles	Accuracy	Vanilla	0.998	0.000557
Circles	F1-score	Full	0.997	0.00309
Circles	F1-score	Vanilla	0.998	0.000558
Linearly Separable	Accuracy	Full	0.999	0.00201
Linearly Separable	Accuracy	Vanilla	1	0
Linearly Separable	F1-score	Full	0.999	0.00201
Linearly Separable	F1-score	Vanilla	1	0
Moons	Accuracy	Full	0.999	0.000696
Moons	Accuracy	Vanilla	1	0.00111
Moons	F1-score	Full	0.999	0.000696
Moons	F1-score	Vanilla	1	0.00111
Overlapping	Accuracy	Full	0.915	0.00477
Overlapping	Accuracy	Vanilla	0.917	0.00123
Overlapping	F1-score	Full	0.915	0.00478
Overlapping	F1-score	Vanilla	0.917	0.00124

649 **J.2.2 Plausibility**

650 The results with respect to the plausibility measure are shown in Figure A4 to Figure A7.

651 **J.2.3 Cost**

652 The results with respect to the cost measure are shown in Figure A8 to Figure A11.

653 **J.3 Penalty Strengths**

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

656 Note 3: Training Phase

- Generator: `ecco`, `generic`, `revise`
- Model: `mlp`
- Training Parameters:
 - λ_{adv} : 0.1, 0.25, 1.0

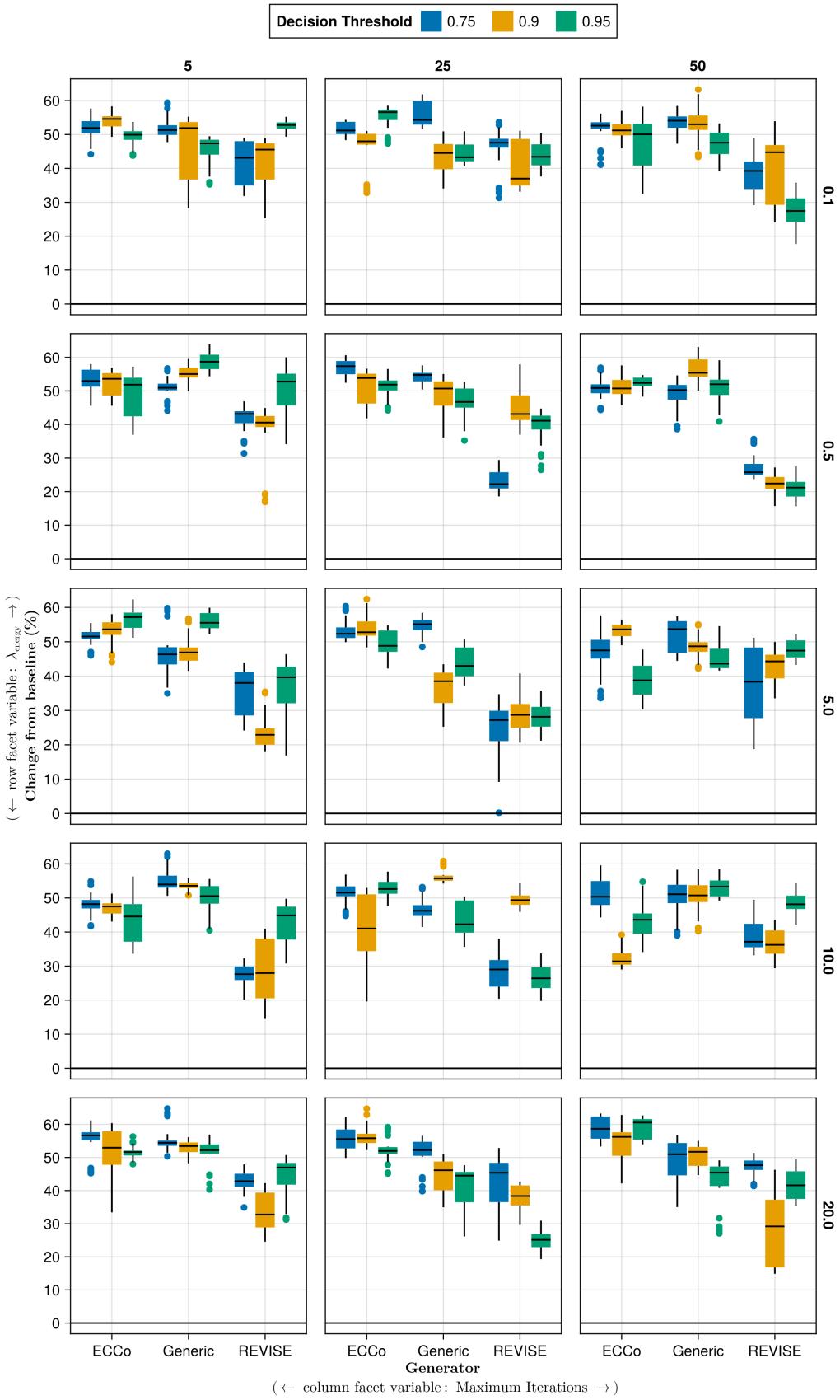


Figure A4: Average outcomes for the plausibility measure across hyperparameters. Data: Circles.

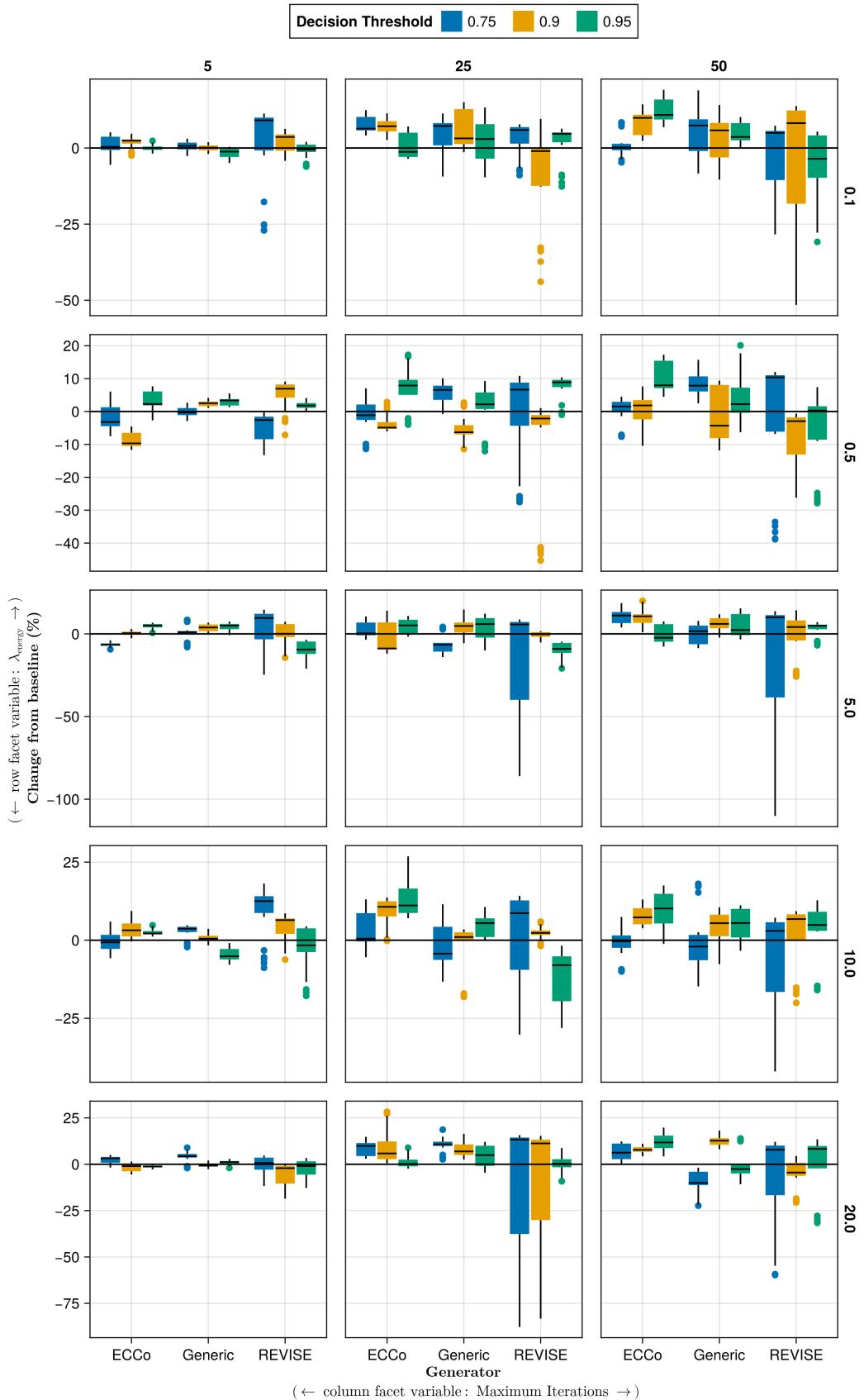


Figure A5: Average outcomes for the plausibility measure across hyperparameters. Data: Linearly Separable.

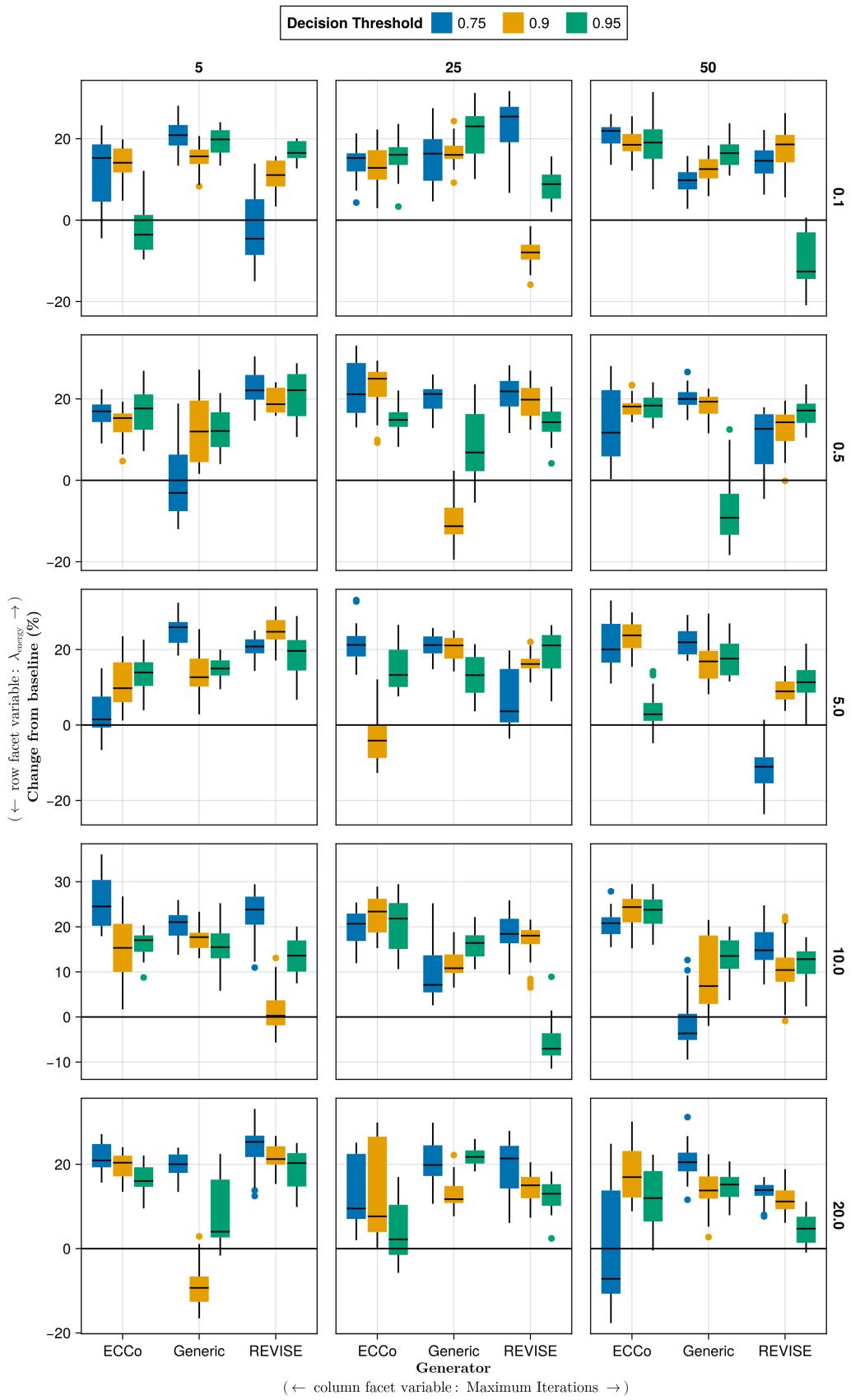


Figure A6: Average outcomes for the plausibility measure across hyperparameters. Data: Moons.

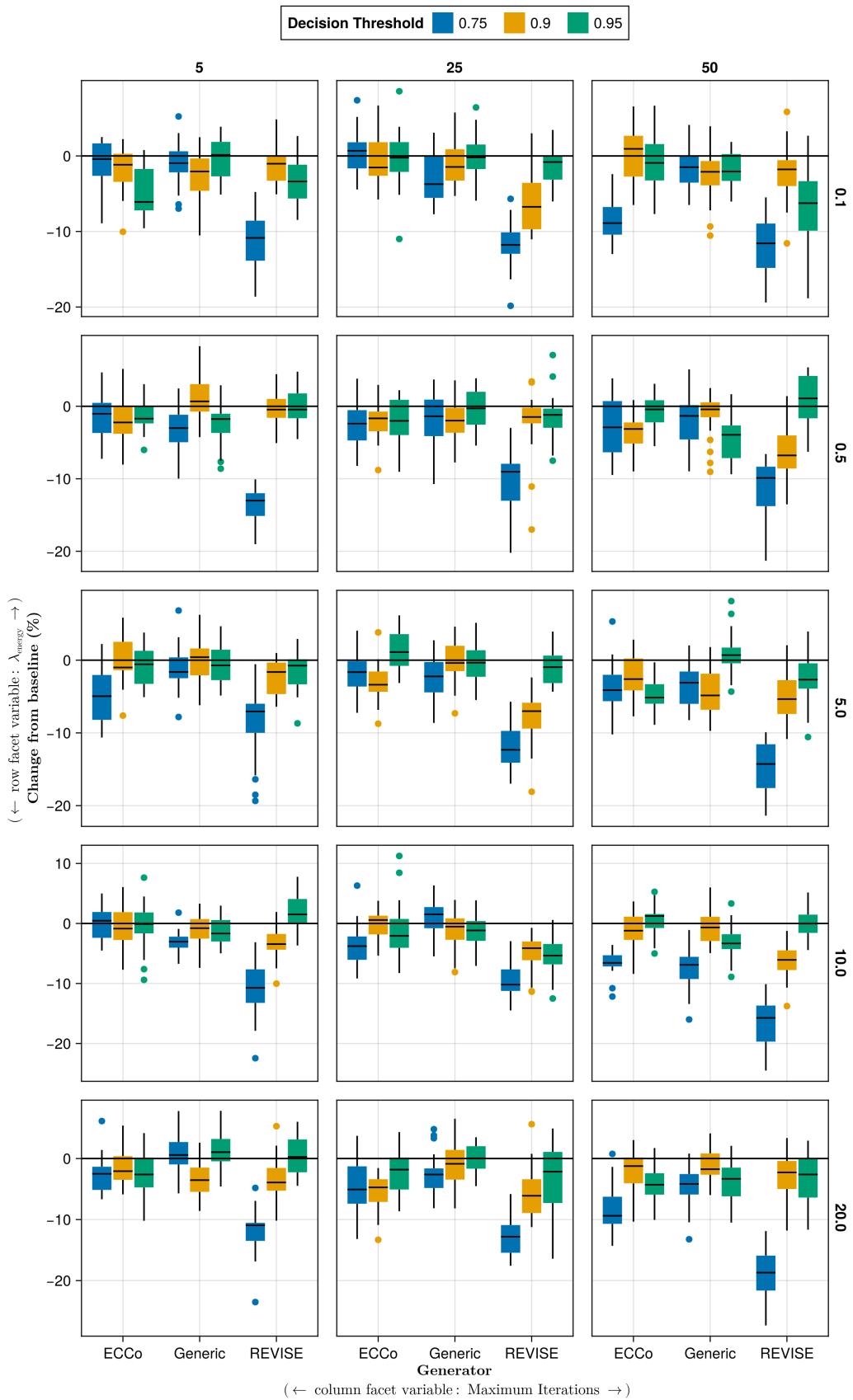


Figure A7: Average outcomes for the plausibility measure across hyperparameters. Data: Overlapping.

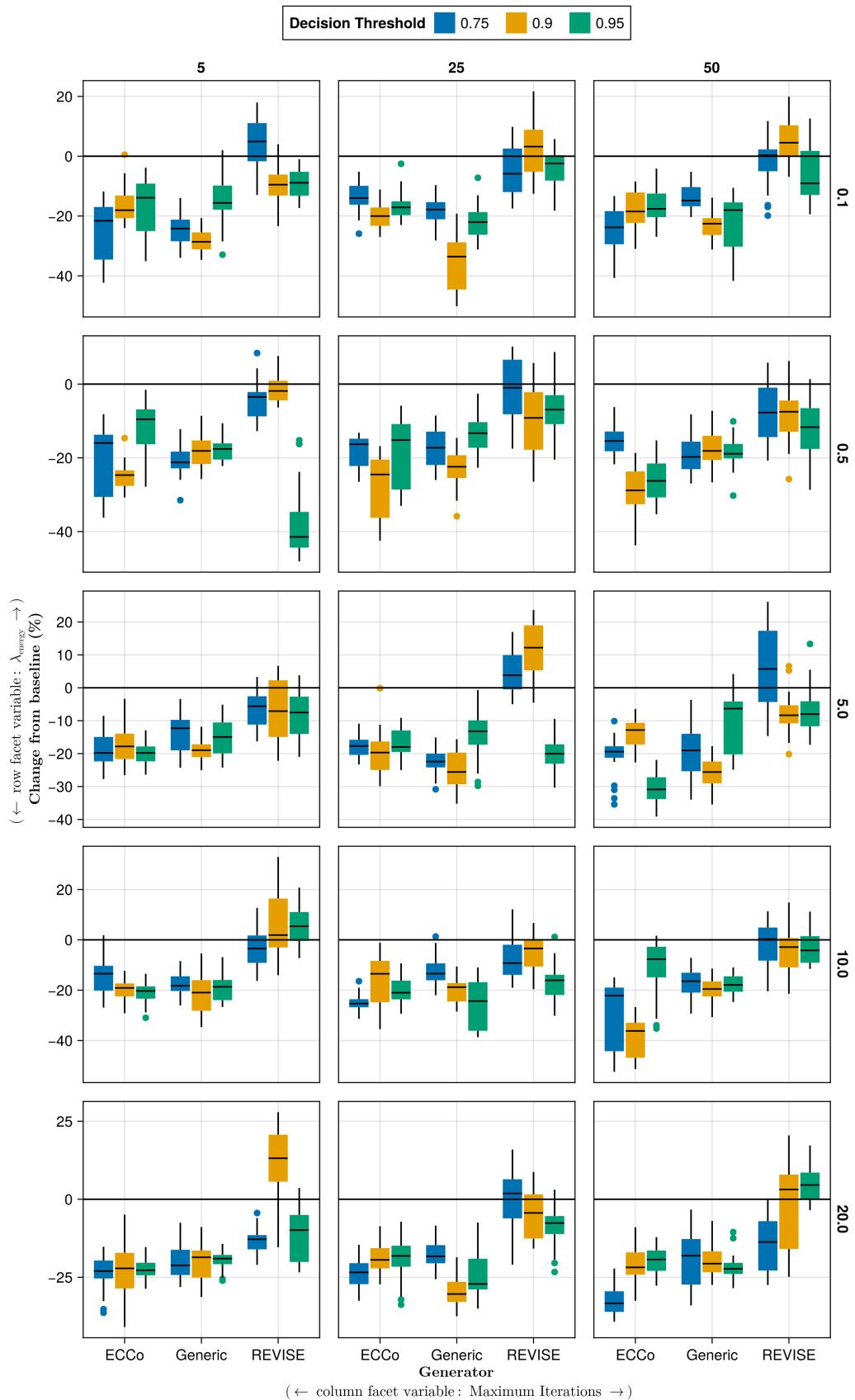


Figure A8: Average outcomes for the cost measure across hyperparameters. Data: Circles.

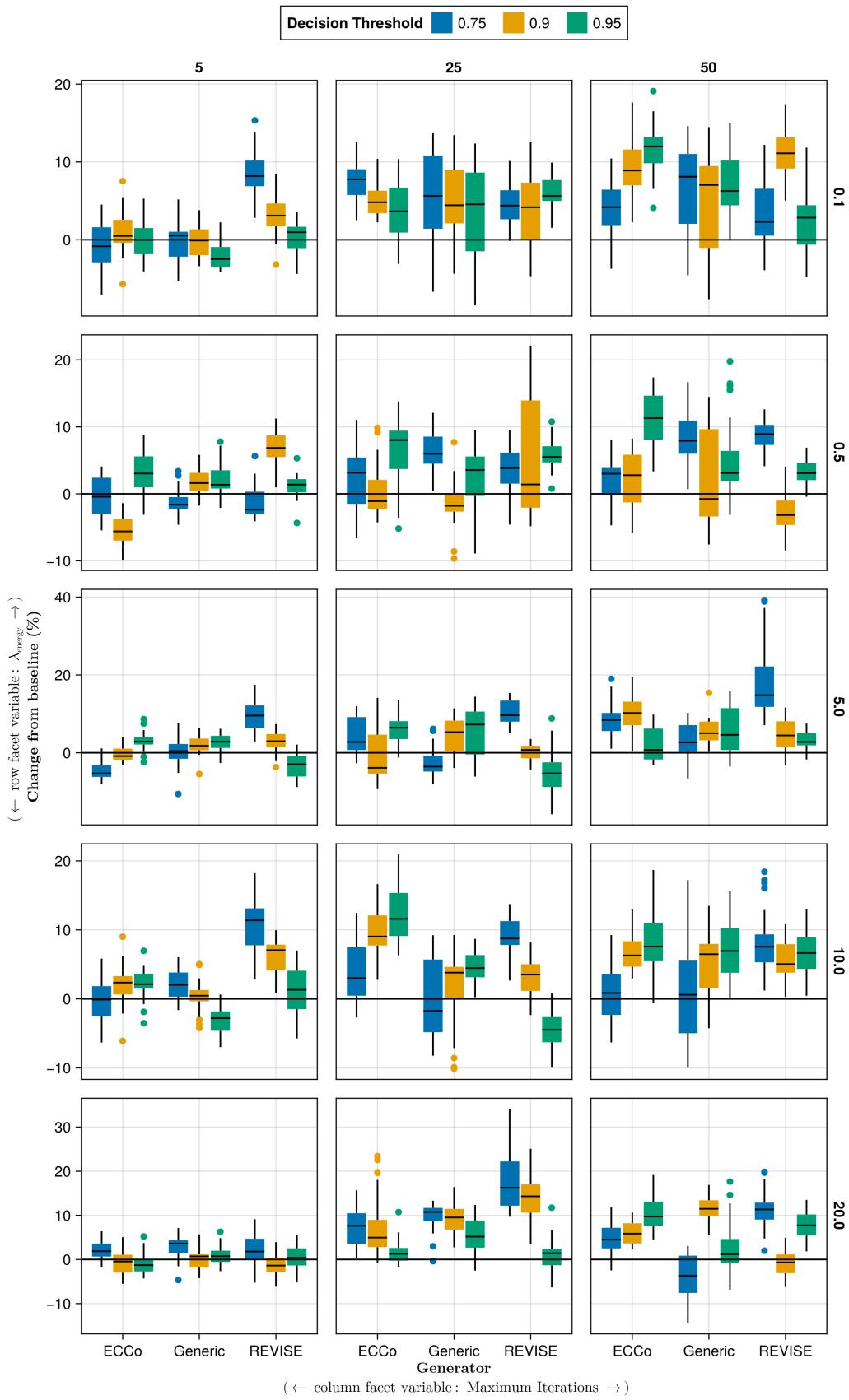


Figure A9: Average outcomes for the cost measure across hyperparameters. Data: Linearly Separable.

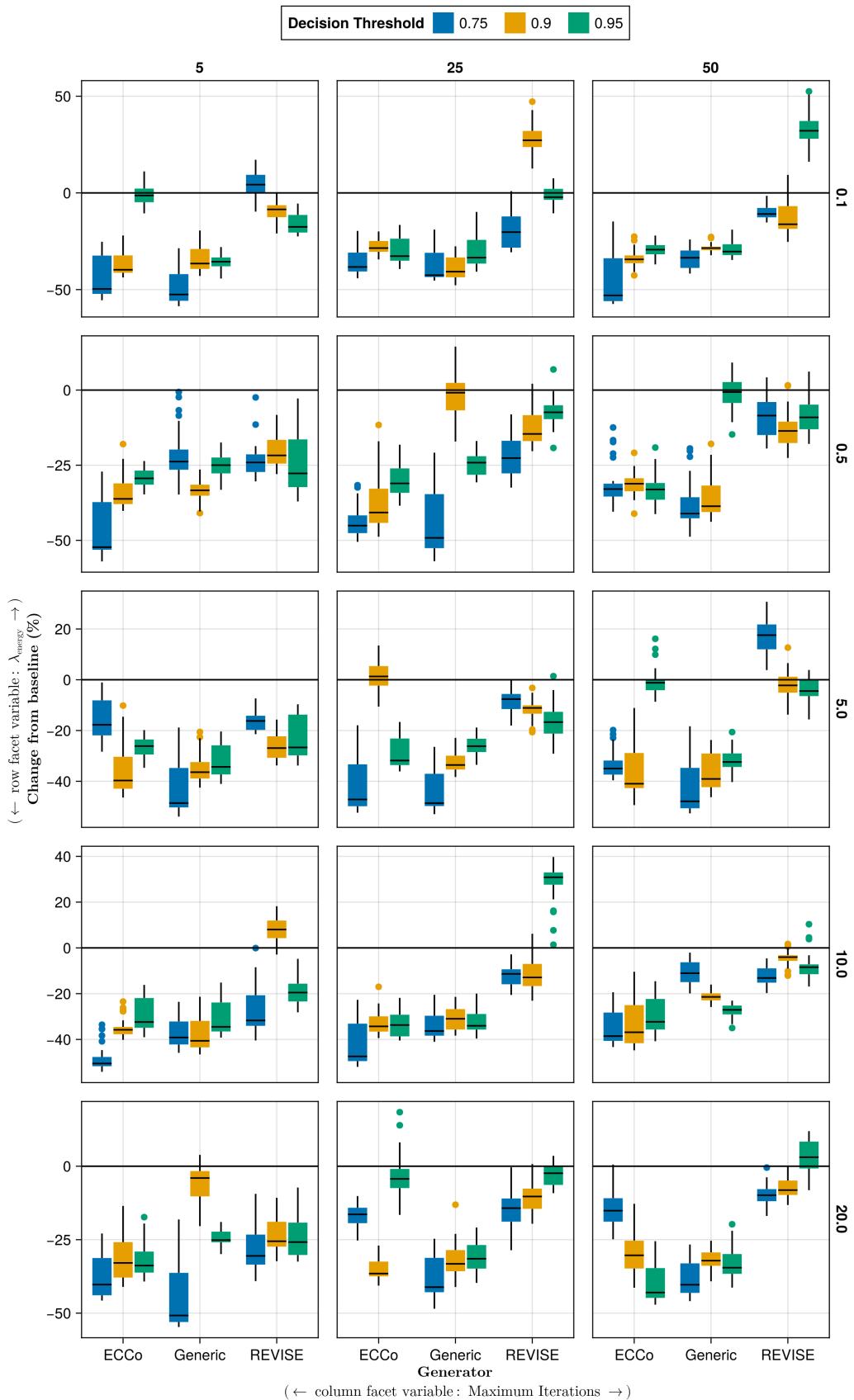


Figure A10: Average outcomes for the cost measure across hyperparameters. Data: Moons.

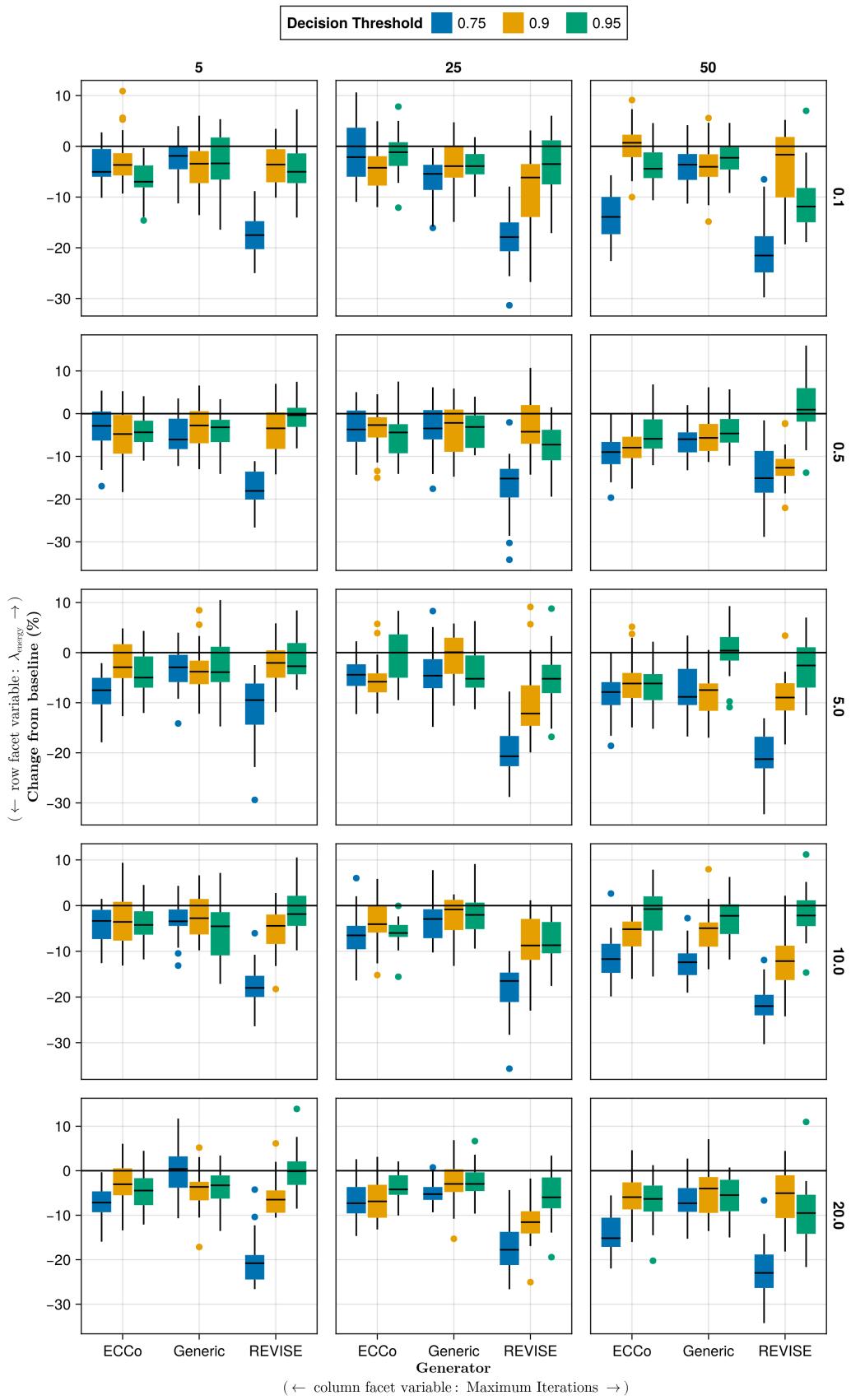


Figure A11: Average outcomes for the cost measure across hyperparameters. Data: Overlapping.

- 657
- λ_{div} : 0.01, 0.1, 1.0
 - λ_{reg} : 0.0, 0.01, 0.1, 0.25, 0.5
 - Objective: full, vanilla

658

Note 4: Evaluation Phase

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

659

J.3.1 Accuracy

Table A3: 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
Circles	Accuracy	Full	0.994	0.0144
Circles	Accuracy	Vanilla	0.998	0.000875
Circles	F1-score	Full	0.994	0.0145
Circles	F1-score	Vanilla	0.998	0.000875
Linearly Separable	Accuracy	Full	0.998	0.00772
Linearly Separable	Accuracy	Vanilla	1	0
Linearly Separable	F1-score	Full	0.998	0.00773
Linearly Separable	F1-score	Vanilla	1	0
Moons	Accuracy	Full	0.987	0.0351
Moons	Accuracy	Vanilla	0.998	0.0101
Moons	F1-score	Full	0.987	0.0352
Moons	F1-score	Vanilla	0.998	0.0102
Overlapping	Accuracy	Full	0.911	0.0217
Overlapping	Accuracy	Vanilla	0.916	0.00236
Overlapping	F1-score	Full	0.911	0.0219
Overlapping	F1-score	Vanilla	0.916	0.00236

660

J.3.2 Plausibility

661 The results with respect to the plausibility measure are shown in Figure A12 to Figure A15.

662

J.3.3 Cost

663 The results with respect to the cost measure are shown in Figure A16 to Figure A19.

664

J.4 Other Parameters

665 The hyperparameter grid with other varying training parameters is shown in Note 5. The corresponding evaluation
666 grid used for these experiments is shown in Note 6.

Note 5: 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

667

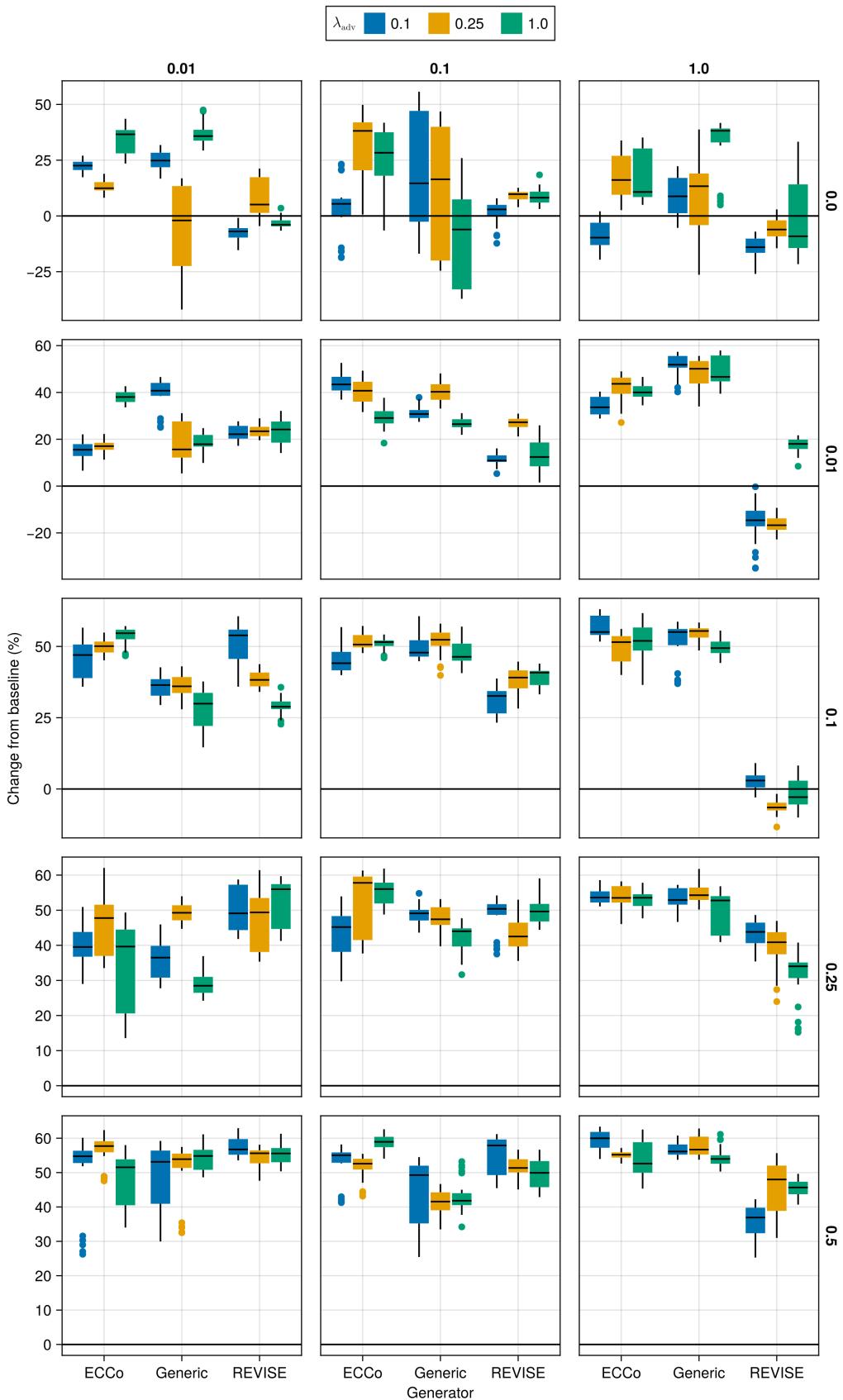


Figure A12: Average outcomes for the plausibility measure across hyperparameters. Data: Circles.

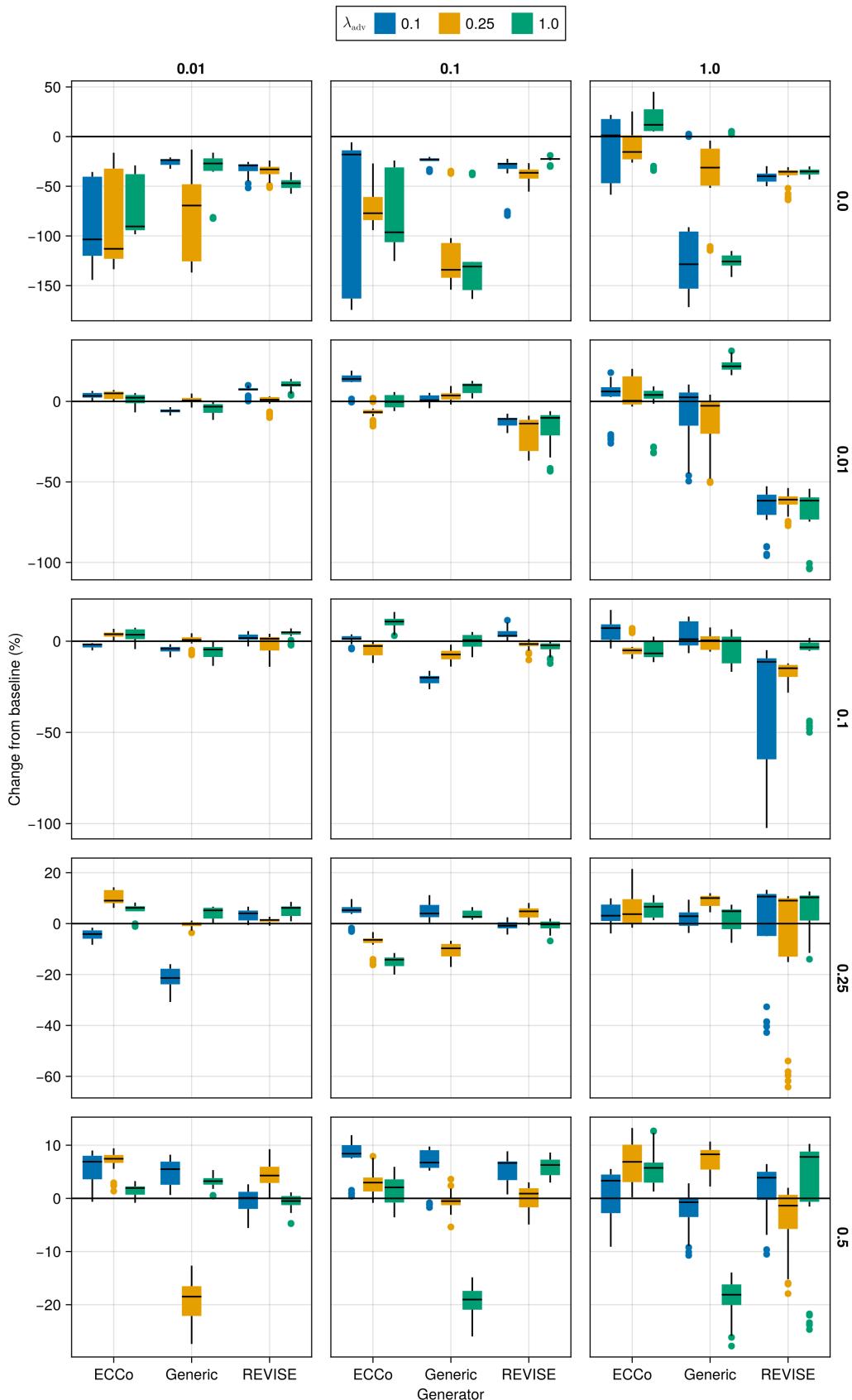


Figure A13: Average outcomes for the plausibility measure across hyperparameters. Data: Linearly Separable.

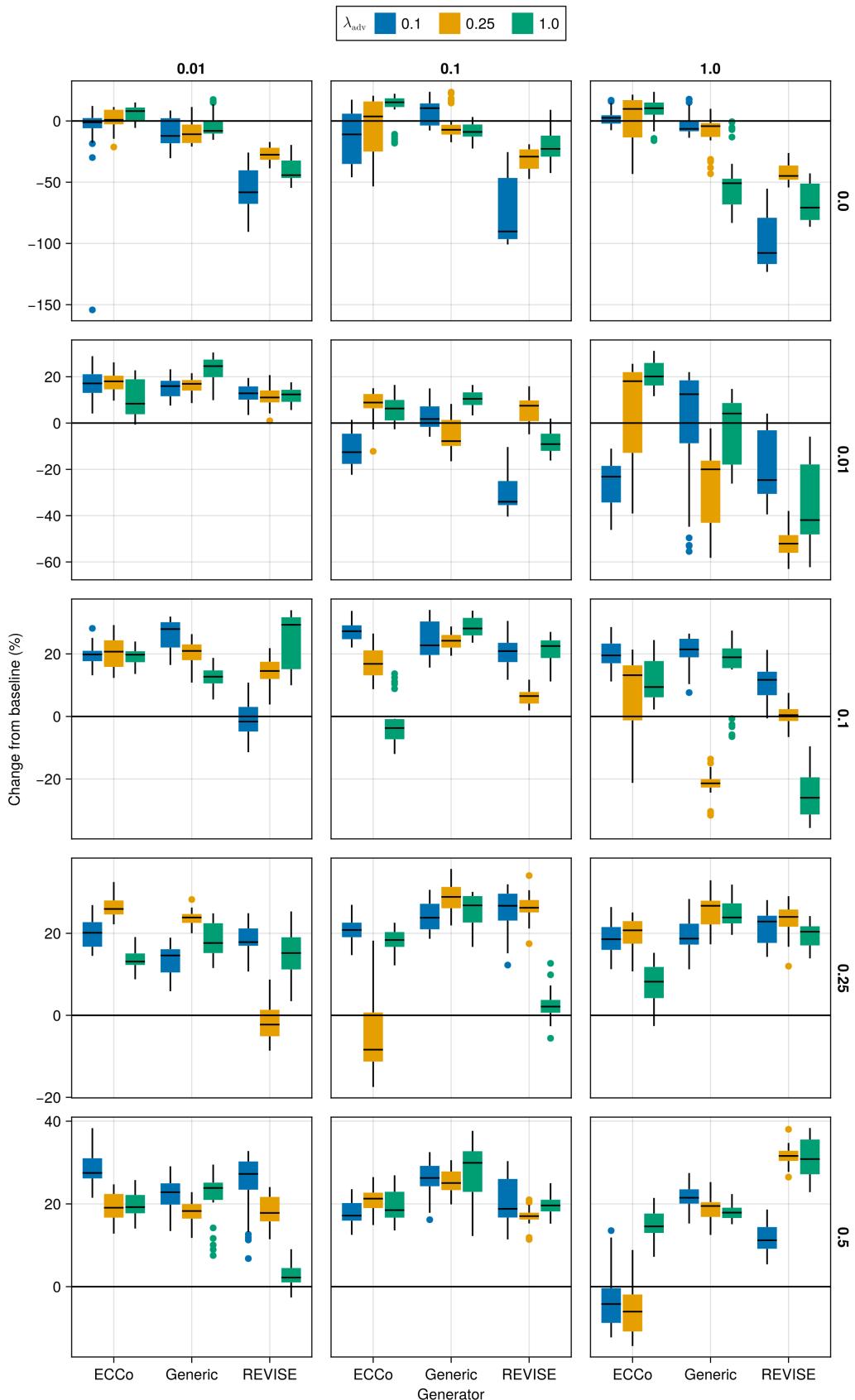


Figure A14: Average outcomes for the plausibility measure across hyperparameters. Data: Moons.

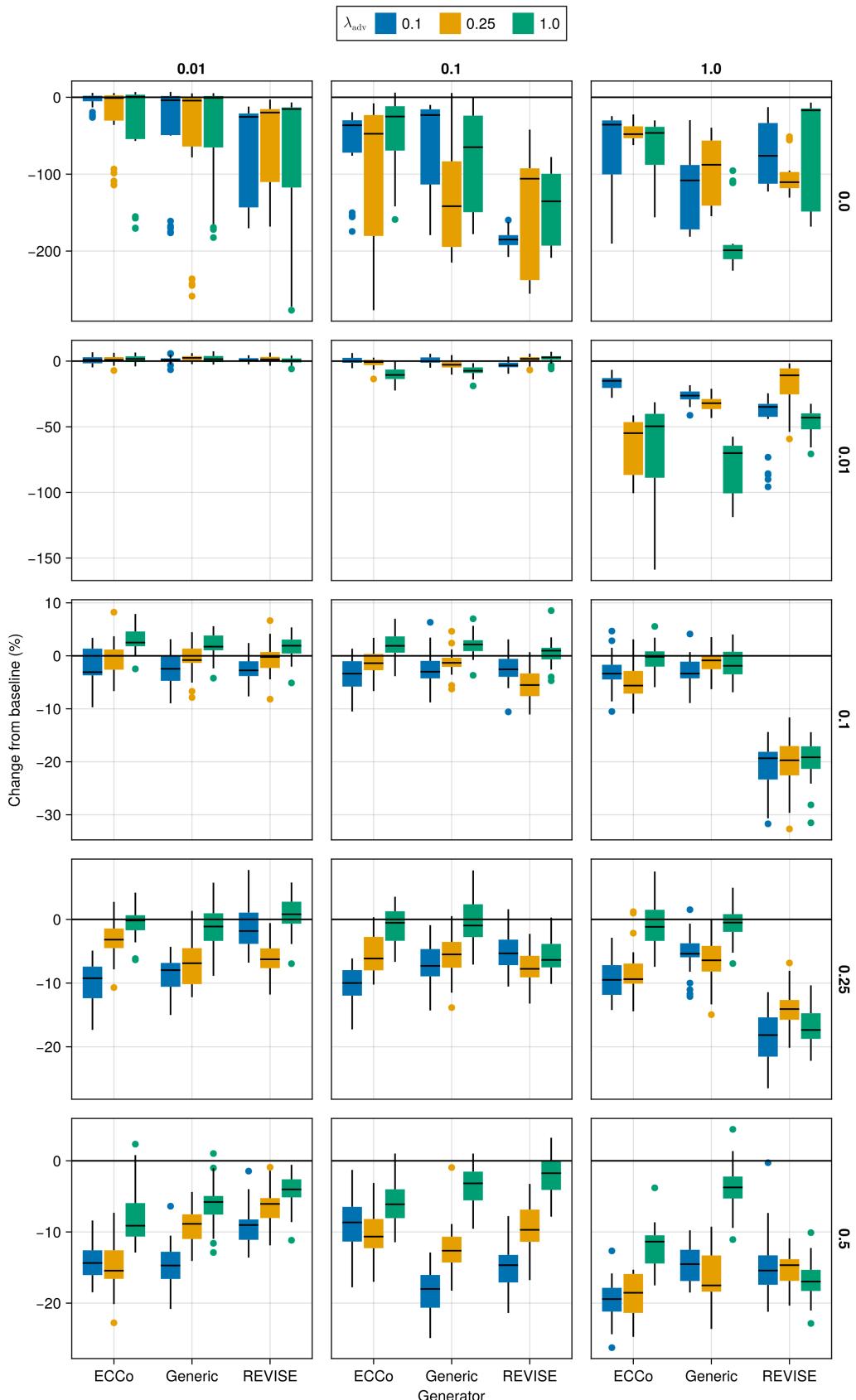


Figure A15: Average outcomes for the plausibility measure across hyperparameters. Data: Overlapping.

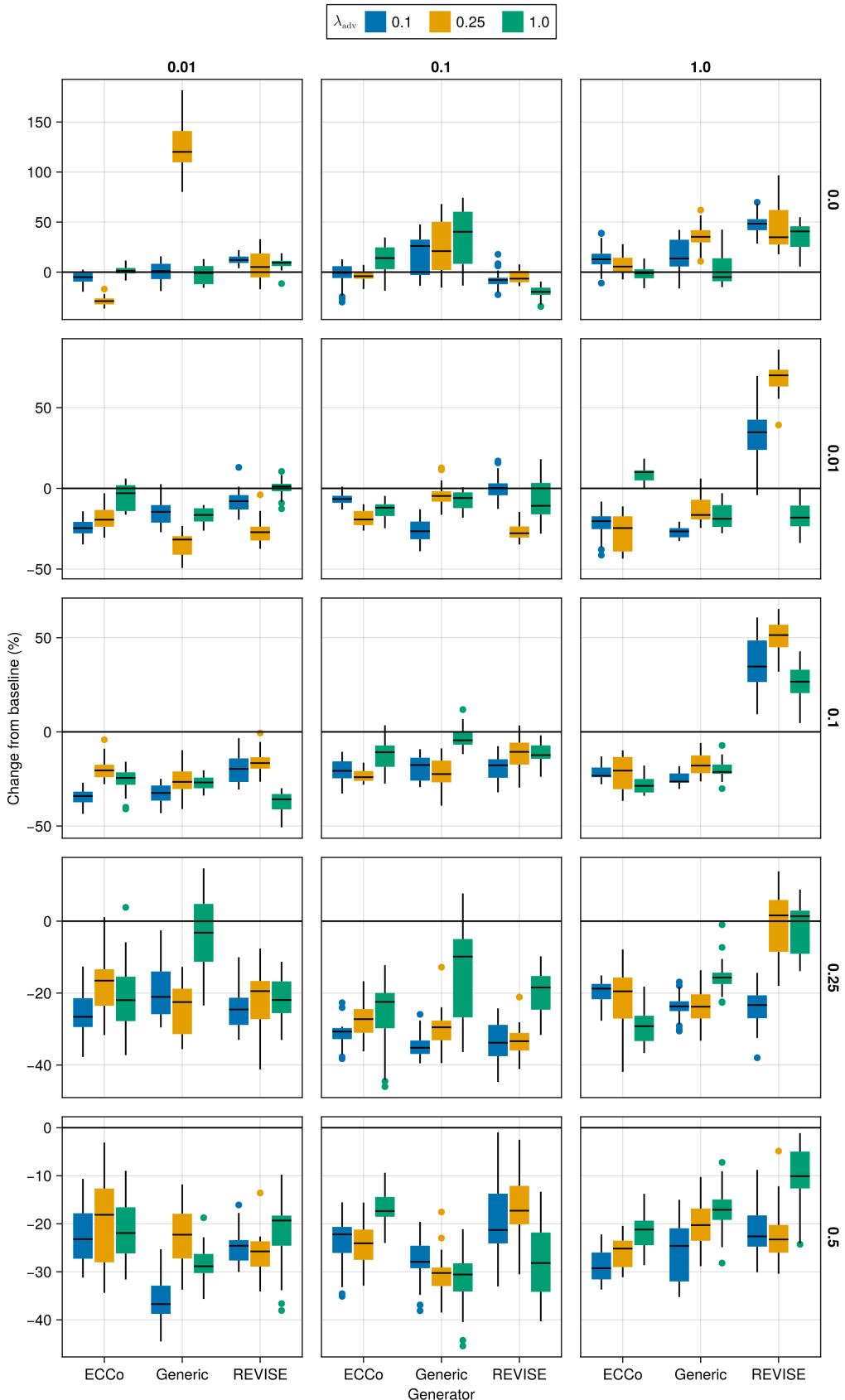


Figure A16: Average outcomes for the cost measure across hyperparameters. Data: Circles.

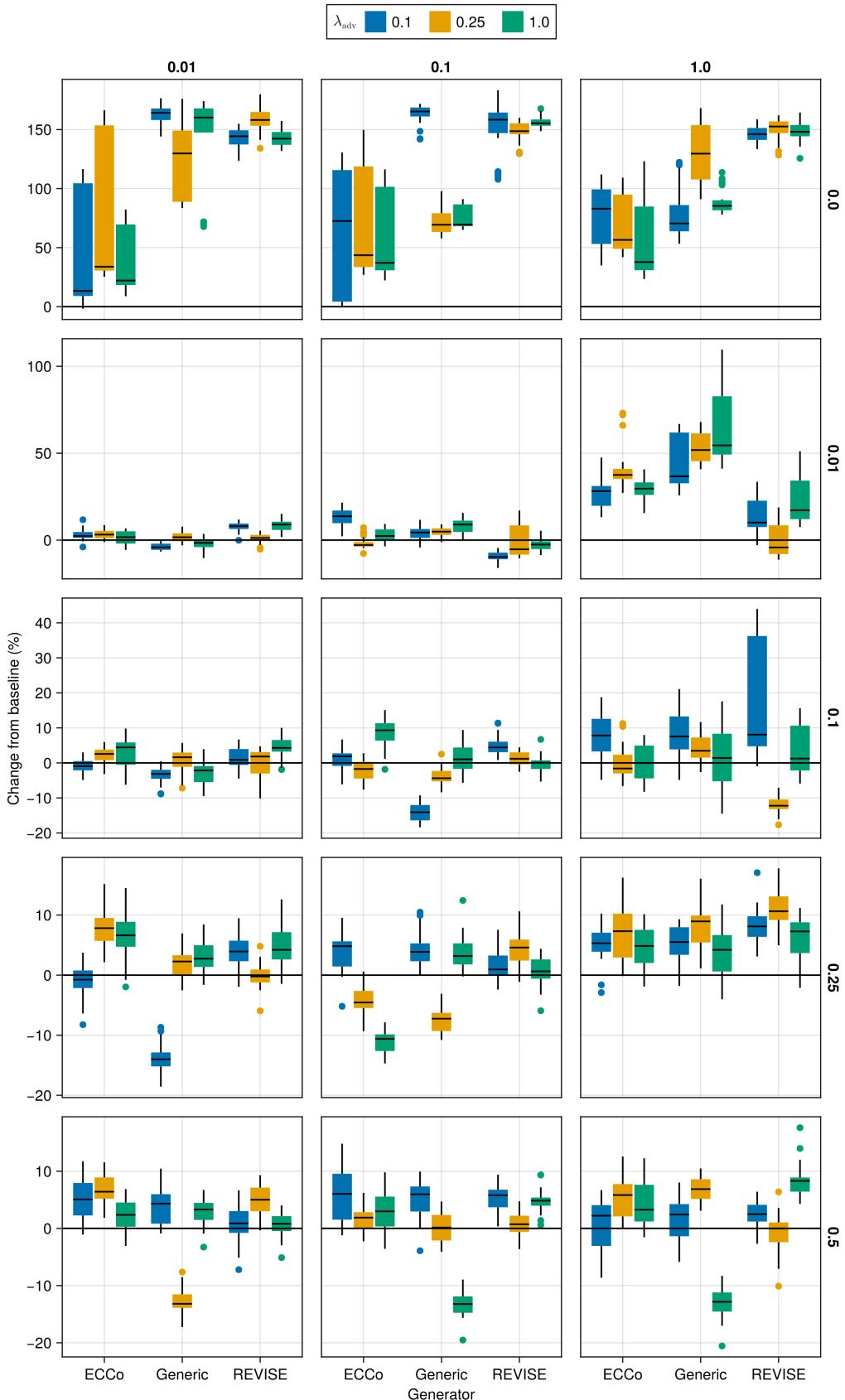


Figure A17: Average outcomes for the cost measure across hyperparameters. Data: Linearly Separable.

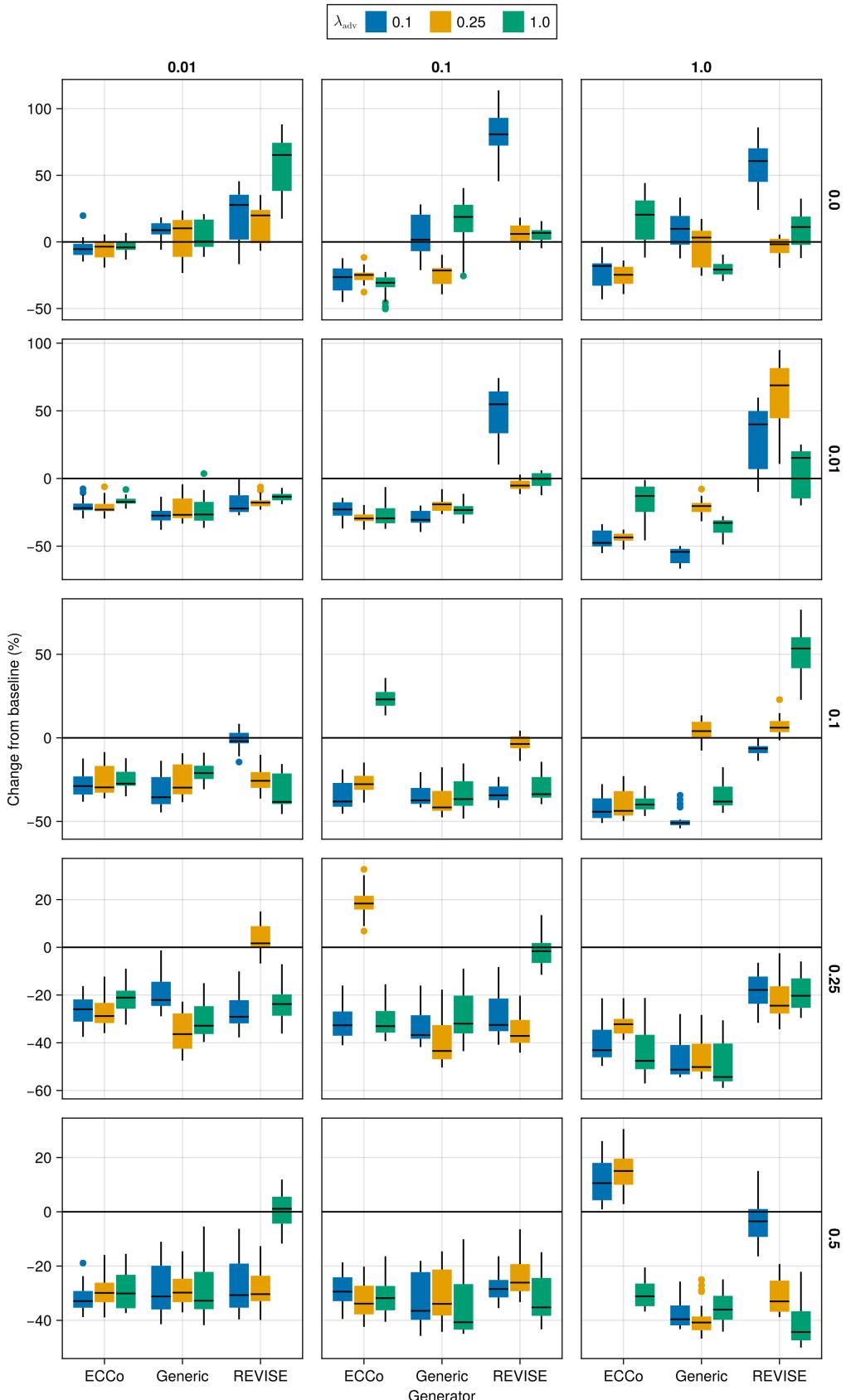


Figure A18: Average outcomes for the cost measure across hyperparameters. Data: Moons.

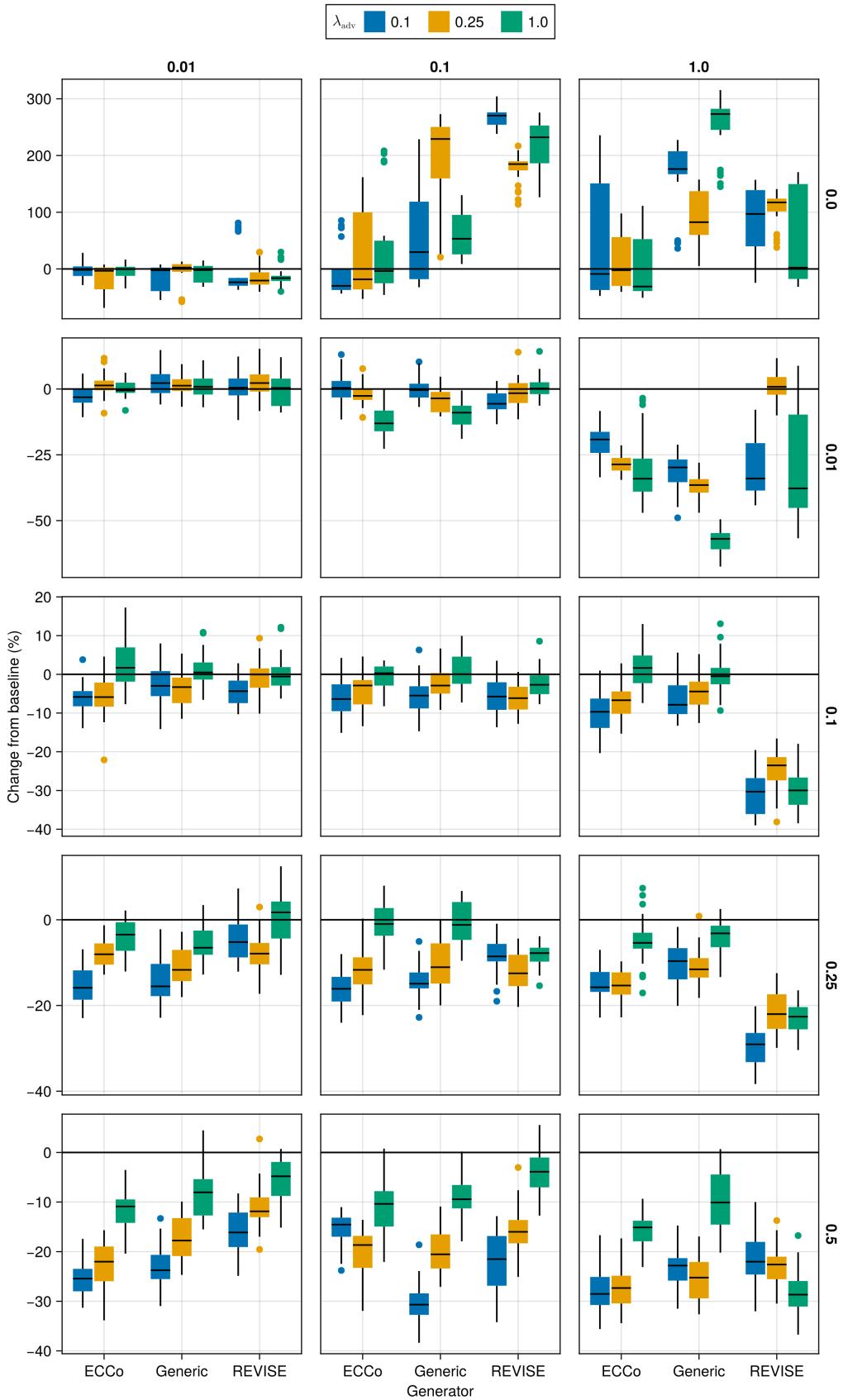


Figure A19: Average outcomes for the cost measure across hyperparameters. Data: Overlapping.

Note 6: Evaluation Phase

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

668

669 **J.4.1 Accuracy**

Table A4: 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
Circles	Accuracy	Full	0.995	0.00431
Circles	Accuracy	Vanilla	0.998	0.000566
Circles	F1-score	Full	0.995	0.00432
Circles	F1-score	Vanilla	0.998	0.000566
Linearly Separable	Accuracy	Full	0.999	0.00231
Linearly Separable	Accuracy	Vanilla	1	0
Linearly Separable	F1-score	Full	0.999	0.00231
Linearly Separable	F1-score	Vanilla	1	0
Moons	Accuracy	Full	0.996	0.0136
Moons	Accuracy	Vanilla	0.988	0.022
Moons	F1-score	Full	0.996	0.0136
Moons	F1-score	Vanilla	0.988	0.022
Overlapping	Accuracy	Full	0.914	0.00563
Overlapping	Accuracy	Vanilla	0.918	0.00116
Overlapping	F1-score	Full	0.914	0.0057
Overlapping	F1-score	Vanilla	0.918	0.00116

670 **J.4.2 Plausibility**

671 The results with respect to the plausibility measure are shown in Figure A20 to Figure A23.

672 **J.4.3 Cost**

673 The results with respect to the cost measure are shown in Figure A24 to Figure A27.

674 **K Tuning Key Parameters**

675 Based on the findings from our initial large grid searches (Section J), we tune selected hyperparameters for all datasets:
 676 namely, the decision threshold τ and the strength of the energy regularization λ_{reg} . The final hyperparameter choices
 677 for each dataset are presented in **ADD TABLE**. Detailed results for each data set are shown in Figure A28 to Fig-
 678 ure A45. From **ADD TABLE**, we notice that the same decision threshold of $\tau = 0.5$ is optimal for all but one dataset.
 679 We attribute this to the fact that a low decision threshold results in a higher share of mature counterfactuals and hence
 680 more opportunities for the model to learn from examples (Figure A37 to Figure A45). This has played a role in par-
 681 ticular for our real-world tabular datasets and MNIST, which suffered from low levels of maturity for higher decision
 682 thresholds. In cases where maturity is not an issue, as for *Moons*, higher decision thresholds lead to better outcomes,
 683 which may have to do with the fact that the resulting counterfactuals are more faithful to the model. Concerning the
 684 regularization strength, we find somewhat high variation across datasets. Most notably, we find that relatively low lev-
 685 els of regularization are optimal for MNIST. We hypothesize that this finding may be attributed to the uniform scaling
 686 of all input features (digits).

687 Finally, to increase the proportion of mature counterfactuals for some datasets, we have also investigated the effect
 688 on the learning rate η for the counterfactual search and even smaller regularization strengths for a fixed decision
 689 threshold of 0.5 (Figure A46 to Figure A50). For the given low decision threshold, we find that the learning rate has
 690 no discernable impact on the proportion of mature counterfactuals (Figure A51 to Figure A55). We do notice, however,
 691 that the results for MNIST are much improved when using a low value λ_{reg} , the strength for the energy regularization:
 692 plausibility is increased by up to ~10% (Figure A49) and the proportion of mature counterfactuals reaches 100%.

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

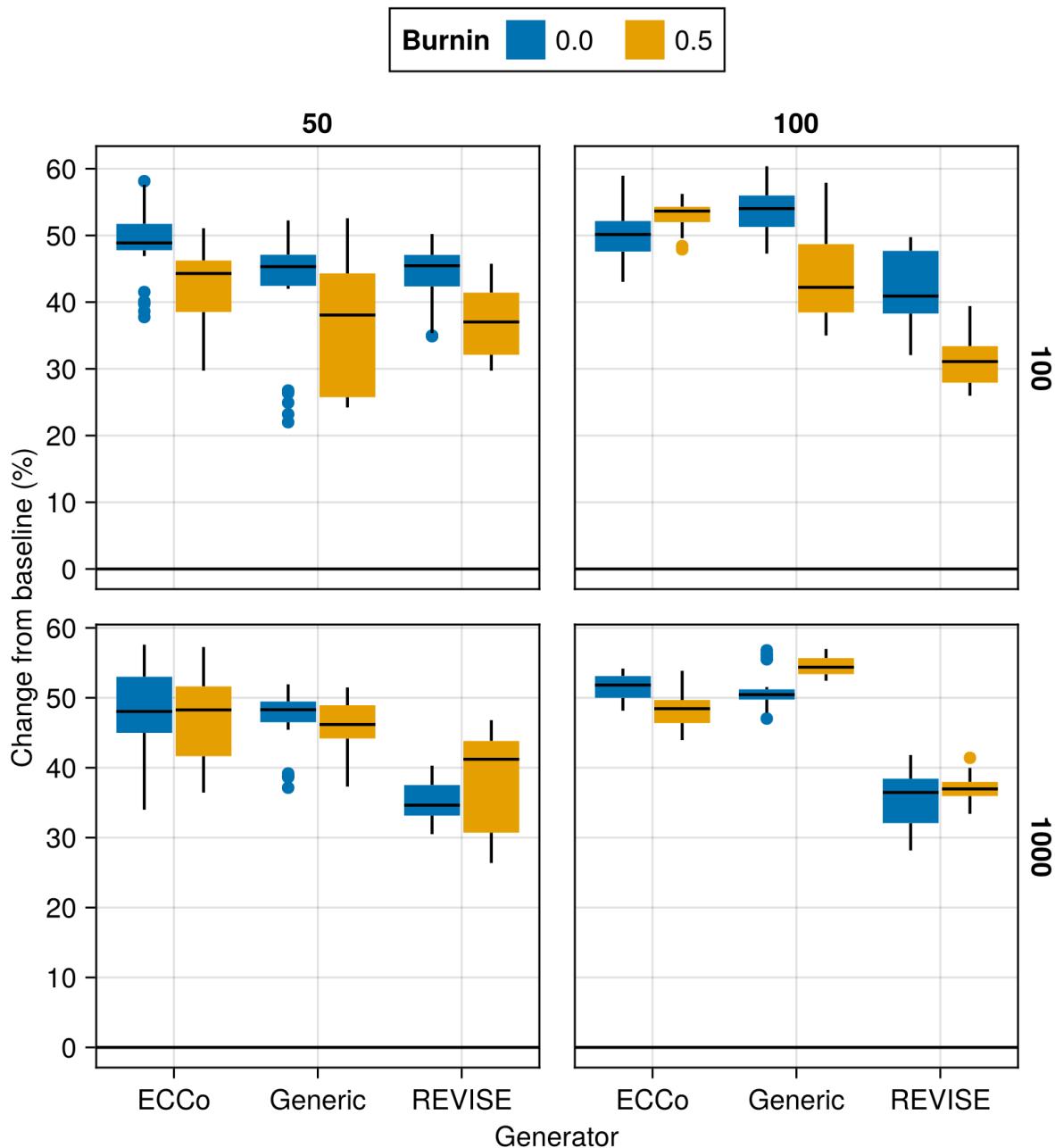


Figure A20: Average outcomes for the plausibility measure across hyperparameters. Data: Circles.

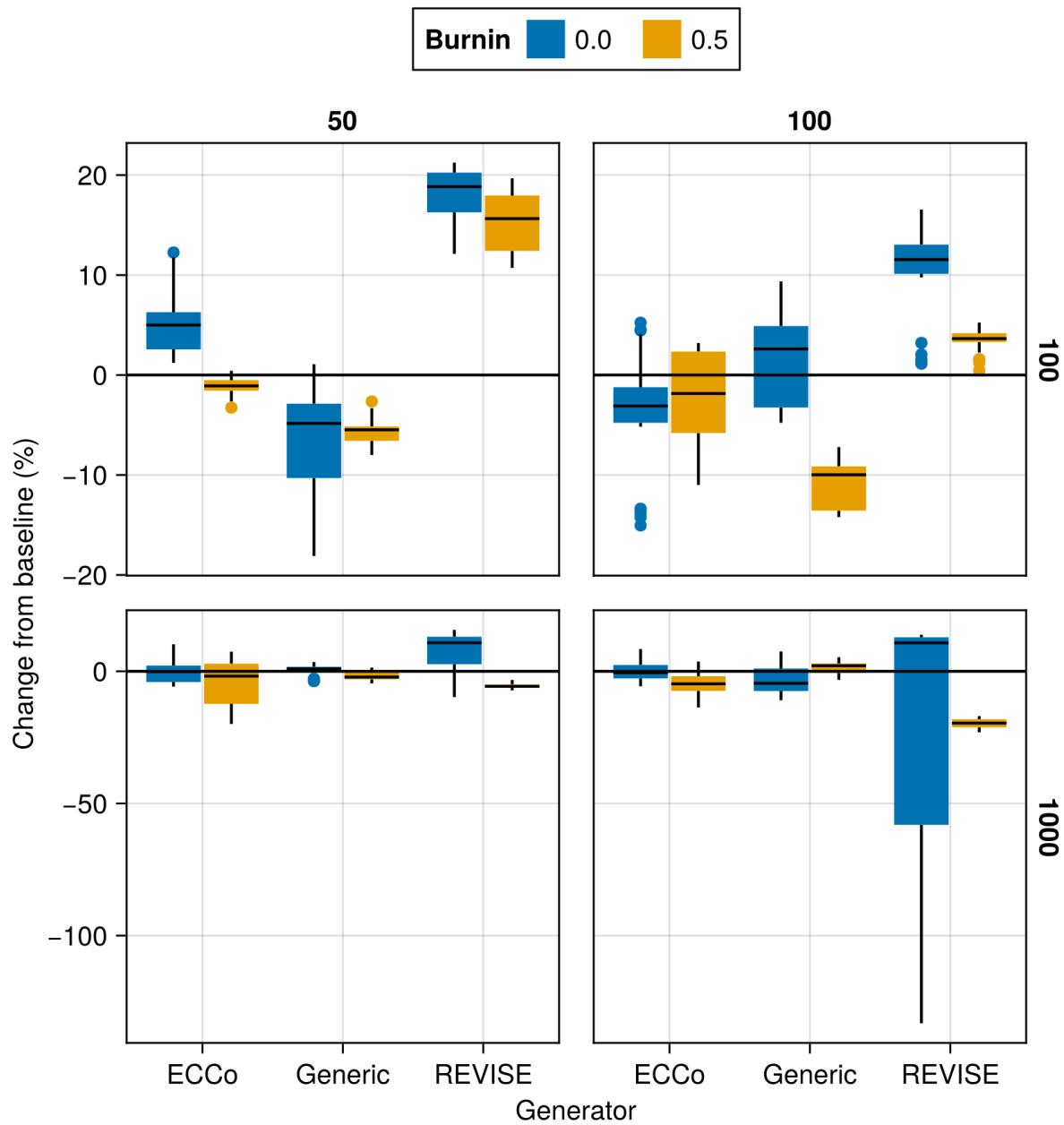


Figure A21: Average outcomes for the plausibility measure across hyperparameters. Data: Linearly Separable.

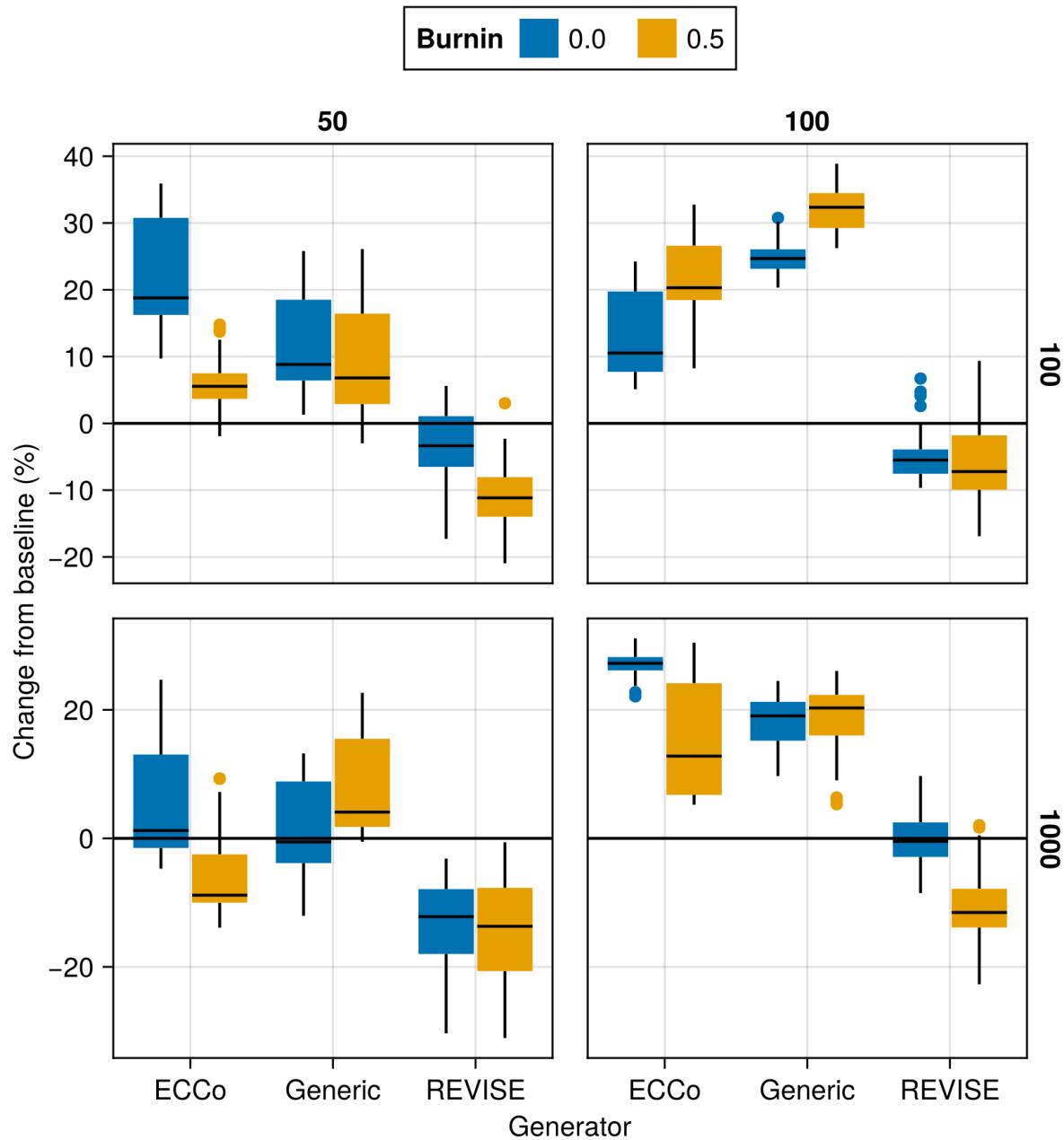


Figure A22: Average outcomes for the plausibility measure across hyperparameters. Data: Moons.

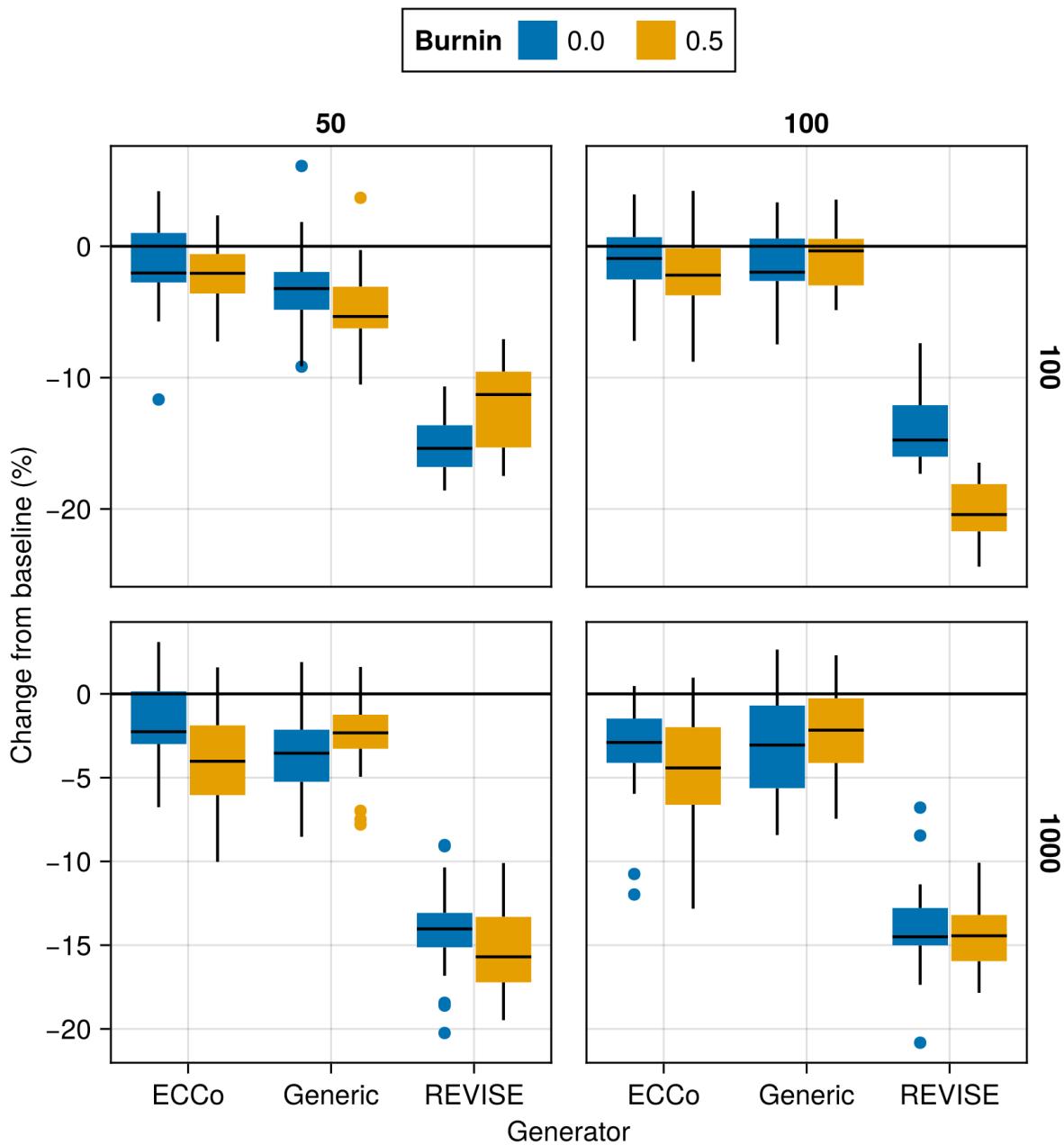


Figure A23: Average outcomes for the plausibility measure across hyperparameters. Data: Overlapping.

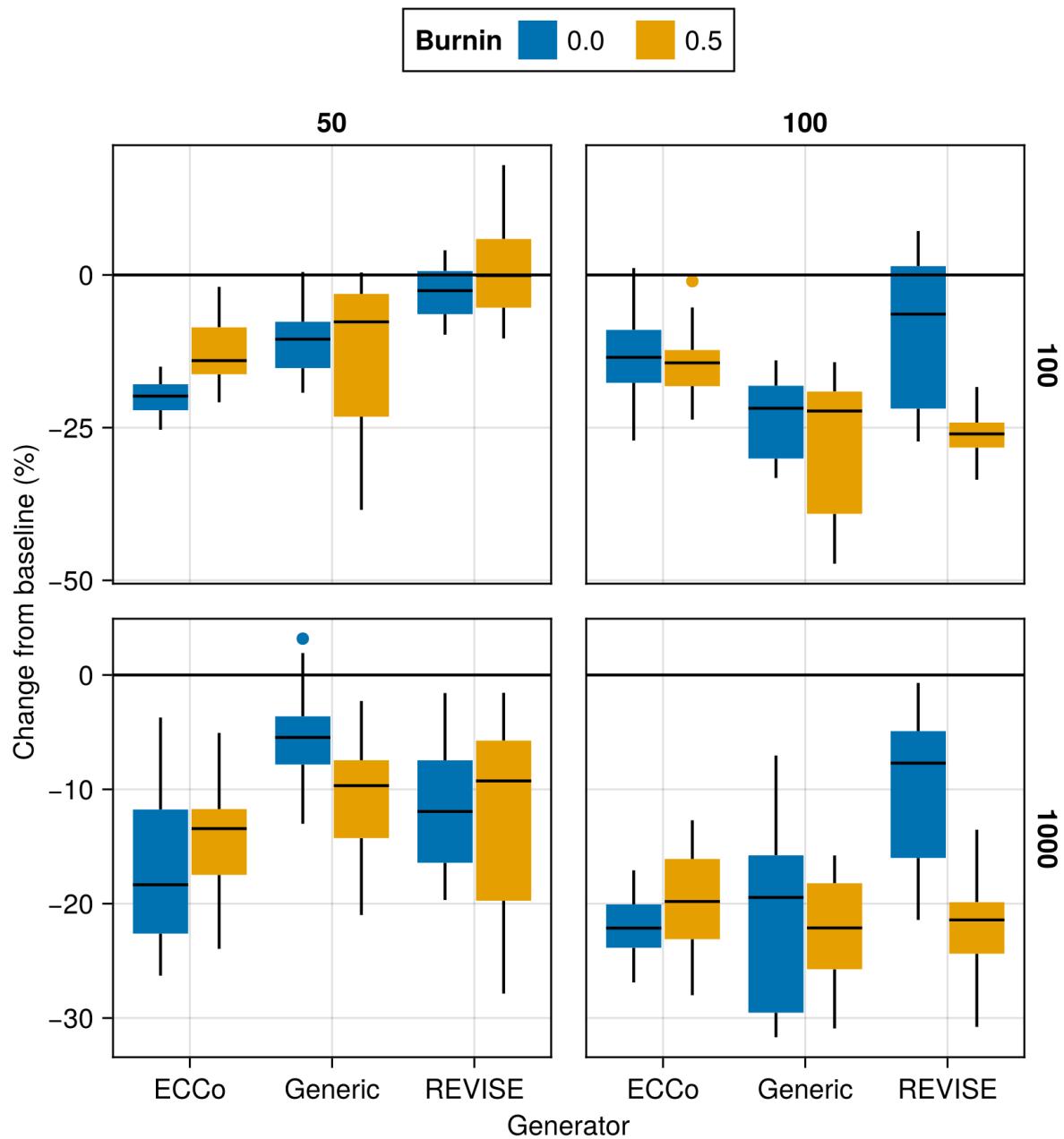


Figure A24: Average outcomes for the cost measure across hyperparameters. Data: Circles.

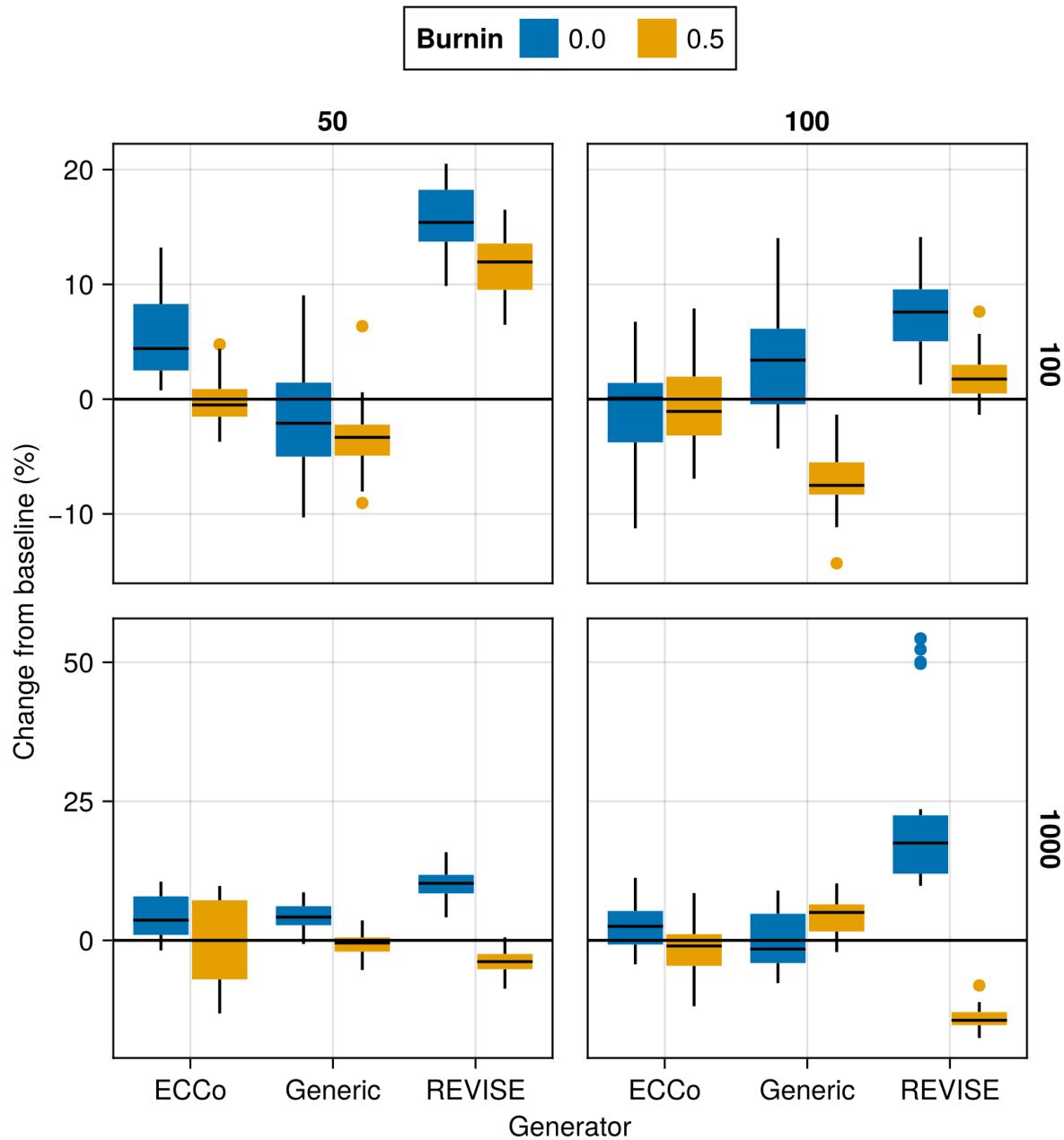


Figure A25: Average outcomes for the cost measure across hyperparameters. Data: Linearly Separable.

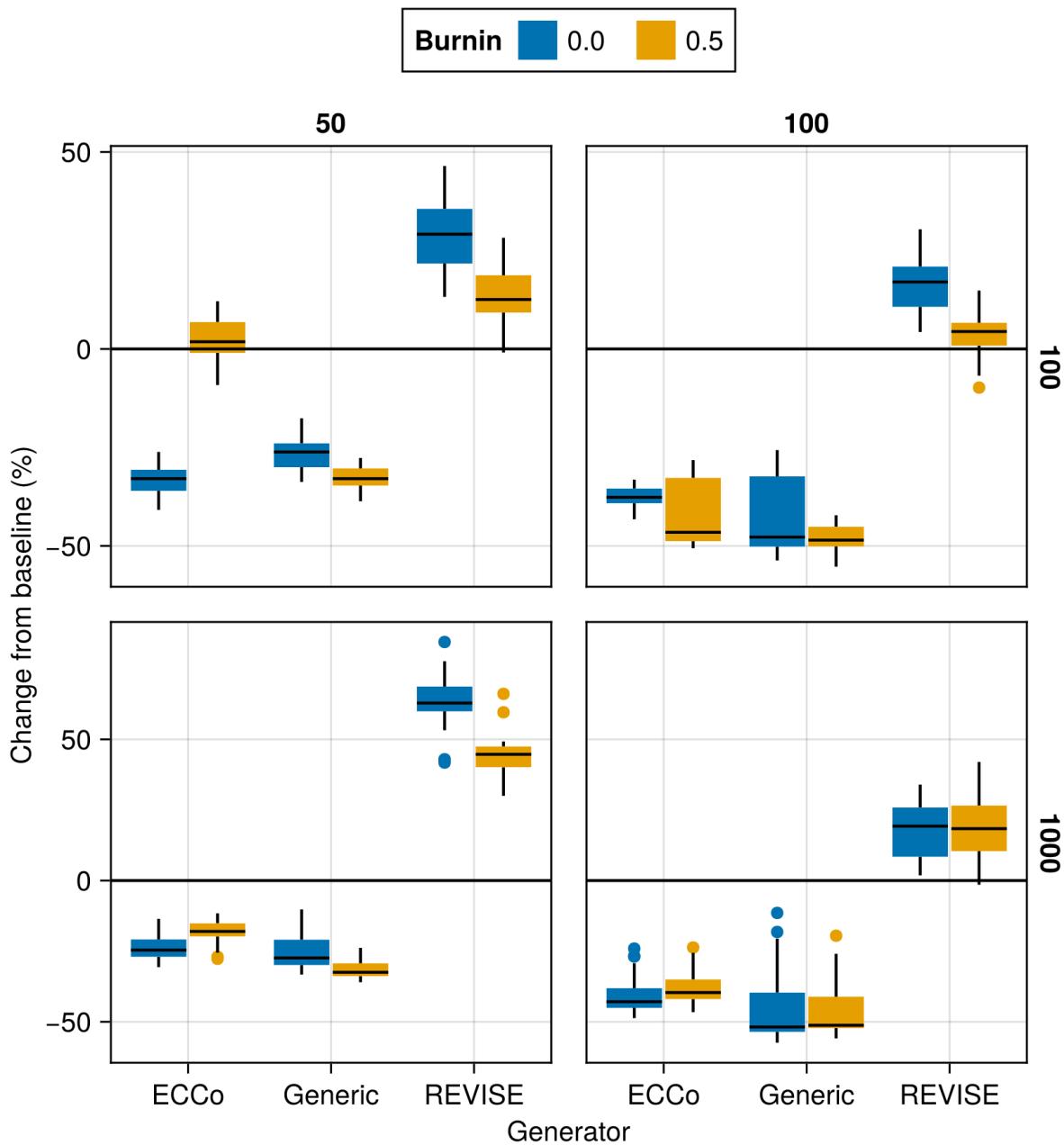


Figure A26: Average outcomes for the cost measure across hyperparameters. Data: Moons.

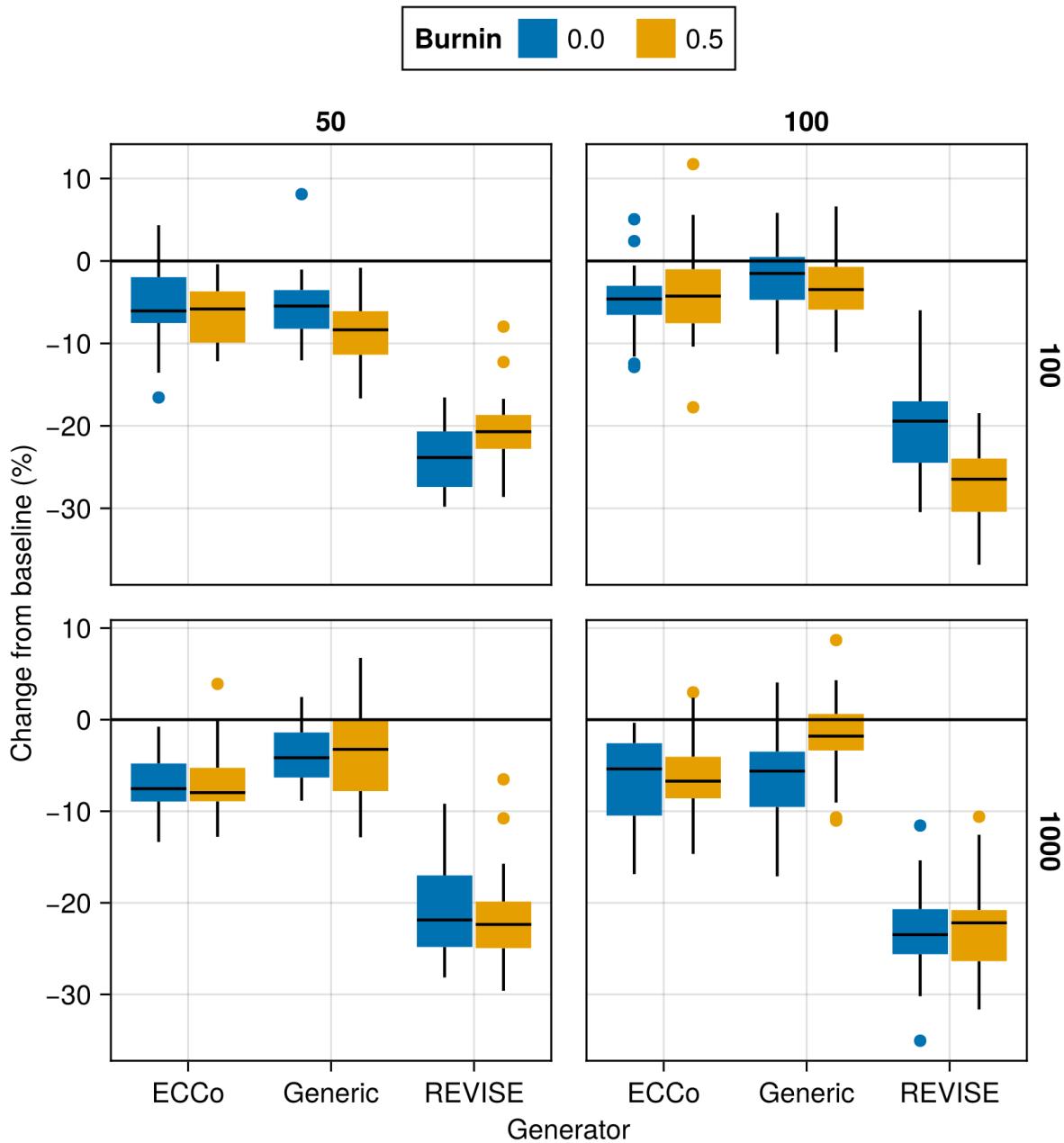


Figure A27: Average outcomes for the cost measure across hyperparameters. Data: Overlapping.

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.

695

696 K.1 Key Parameters

697 The hyperparameter grid for tuning key parameters is shown in Note 7. The corresponding evaluation grid used for
698 these experiments is shown in Note 8.

Note 7: Training Phase

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

699

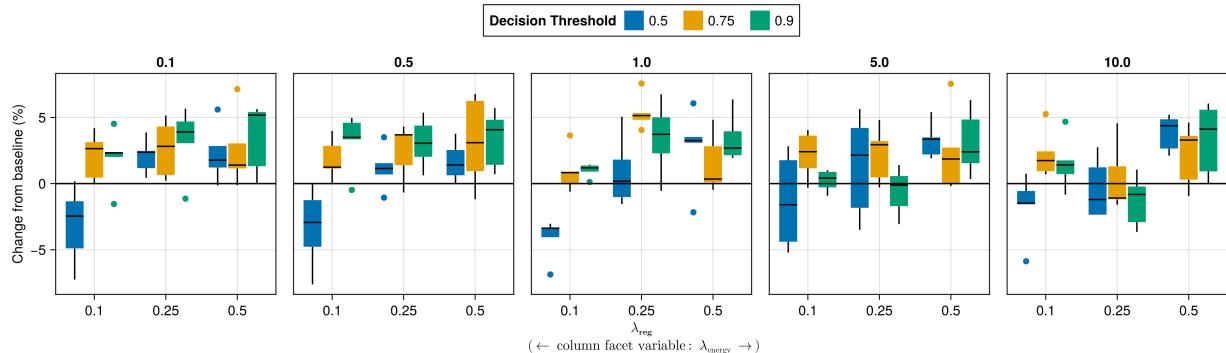
Note 8: Evaluation Phase

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

700

K.1.1 Plausibility

701 The results with respect to the plausibility measure are shown in Figure A28 to Figure A36.



702 Figure A28: Average outcomes for the plausibility measure across key hyperparameters. Data: Adult.

703 K.1.2 Proportion of Mature CE

704 The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure A37 to Figure
705 A45.

706 K.2 Learning Rate

707 The hyperparameter grid for tuning the learning rate is shown in Note 9. The corresponding evaluation grid used for
708 these experiments is shown in Note 10.

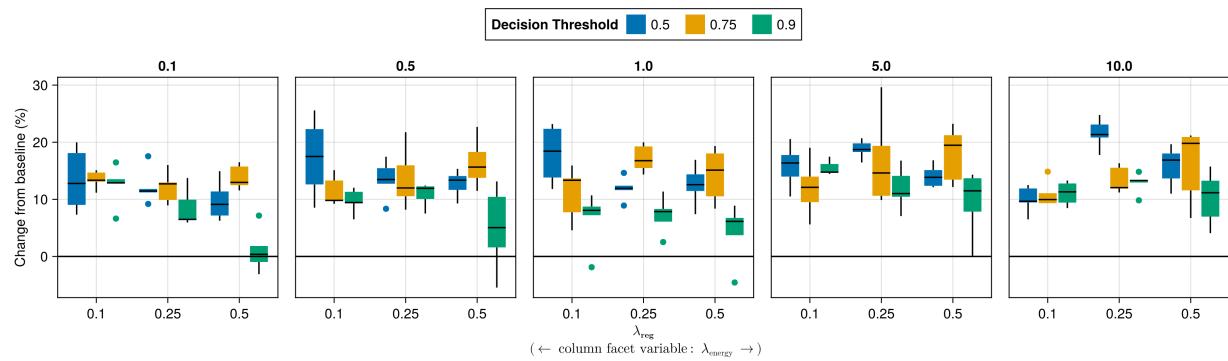


Figure A29: Average outcomes for the plausibility measure across key hyperparameters. Data: California Housing.

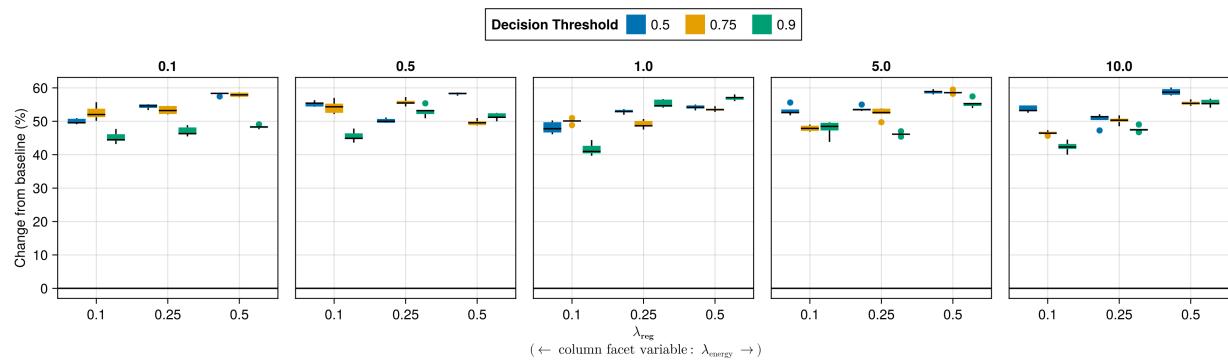


Figure A30: Average outcomes for the plausibility measure across key hyperparameters. Data: Circles.

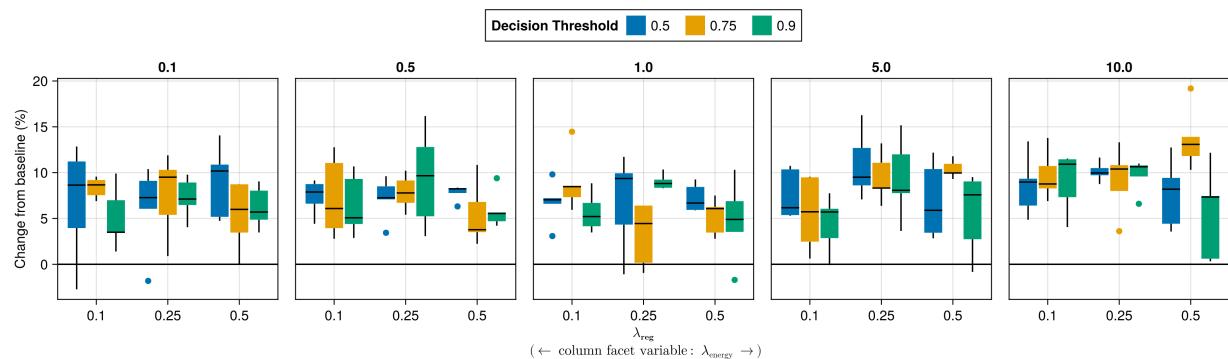


Figure A31: Average outcomes for the plausibility measure across key hyperparameters. Data: Credit.

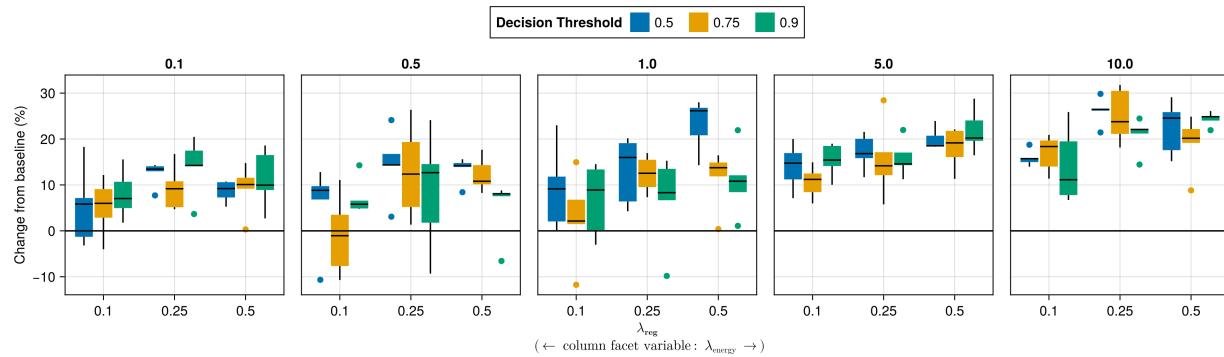


Figure A32: Average outcomes for the plausibility measure across key hyperparameters. Data: GMSC.

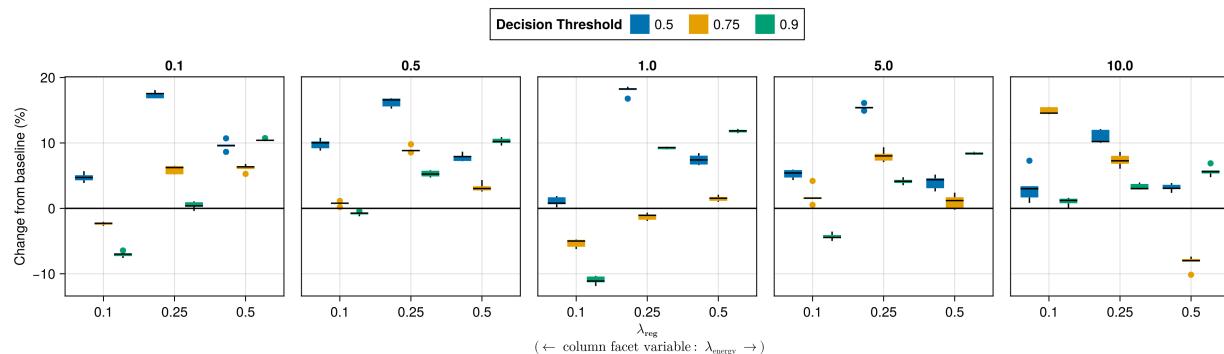


Figure A33: Average outcomes for the plausibility measure across key hyperparameters. Data: Linearly Separable.

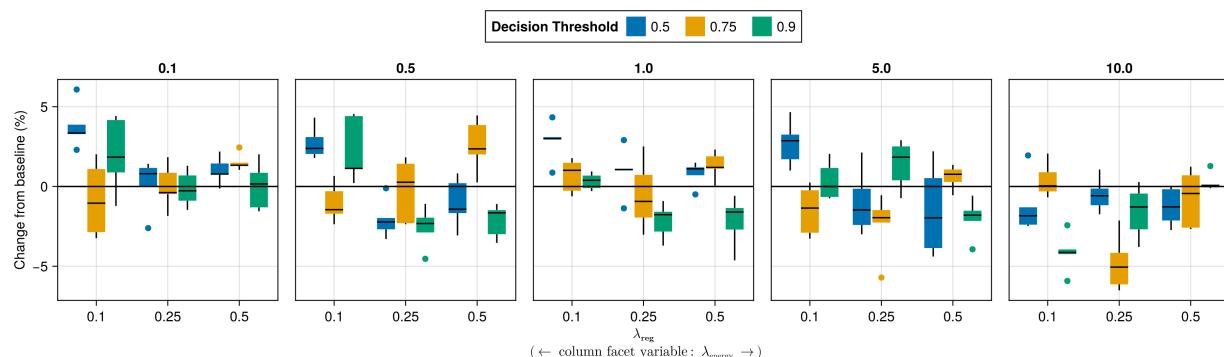


Figure A34: Average outcomes for the plausibility measure across key hyperparameters. Data: MNIST.

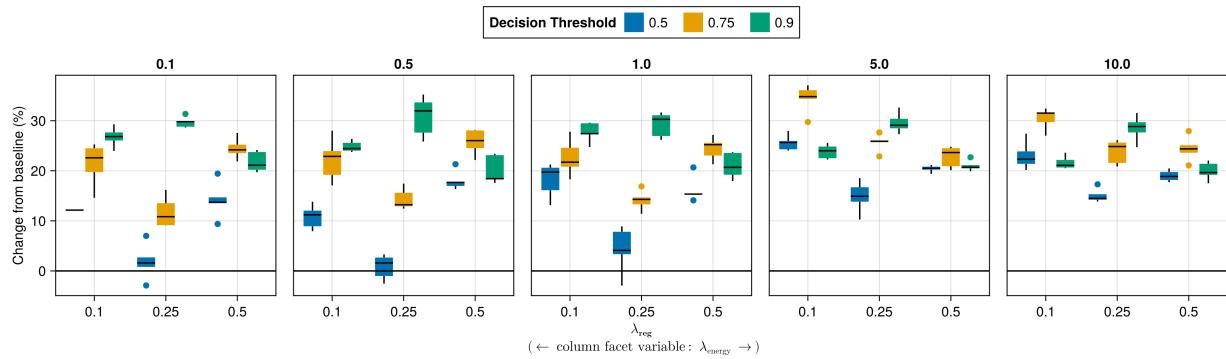


Figure A35: Average outcomes for the plausibility measure across key hyperparameters. Data: Moons.

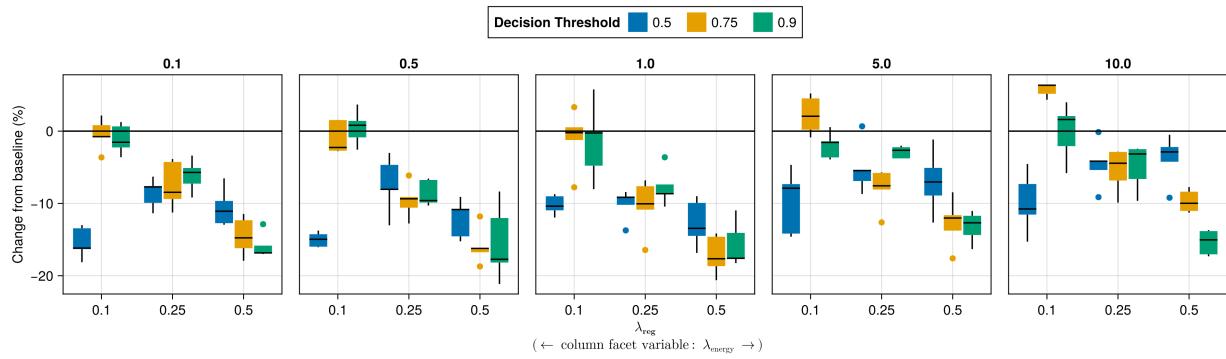


Figure A36: Average outcomes for the plausibility measure across key hyperparameters. Data: Overlapping.

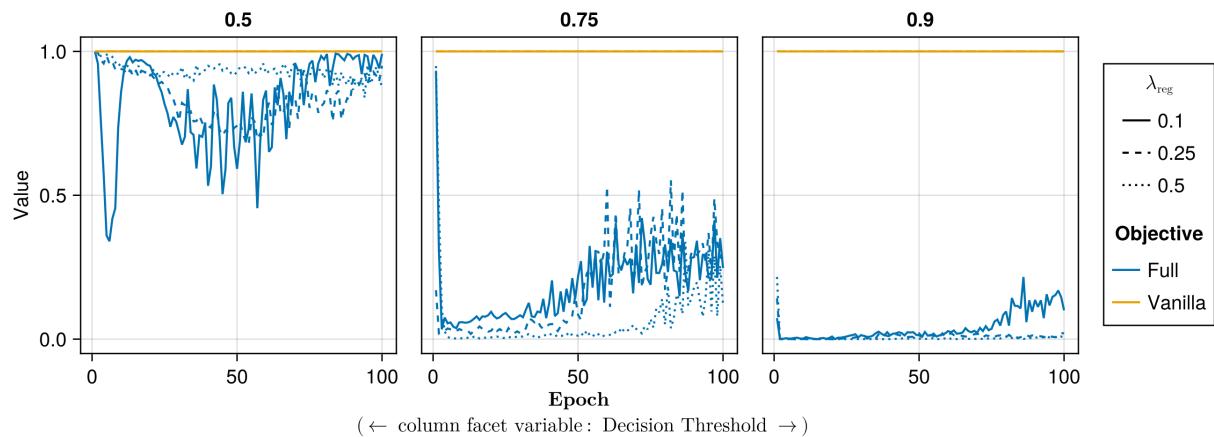


Figure A37: Proportion of mature counterfactuals in each epoch. Data: Adult.

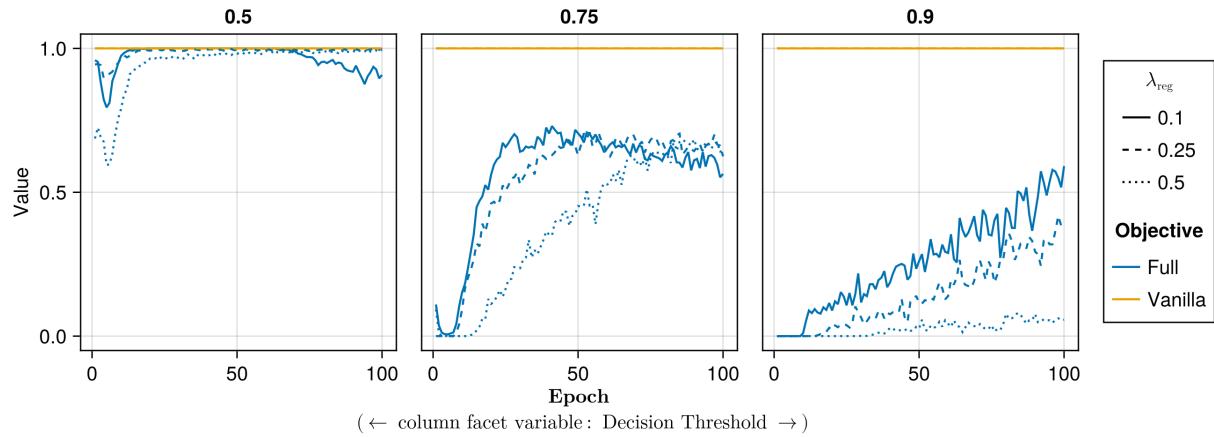


Figure A38: Proportion of mature counterfactuals in each epoch. Data: California Housing.

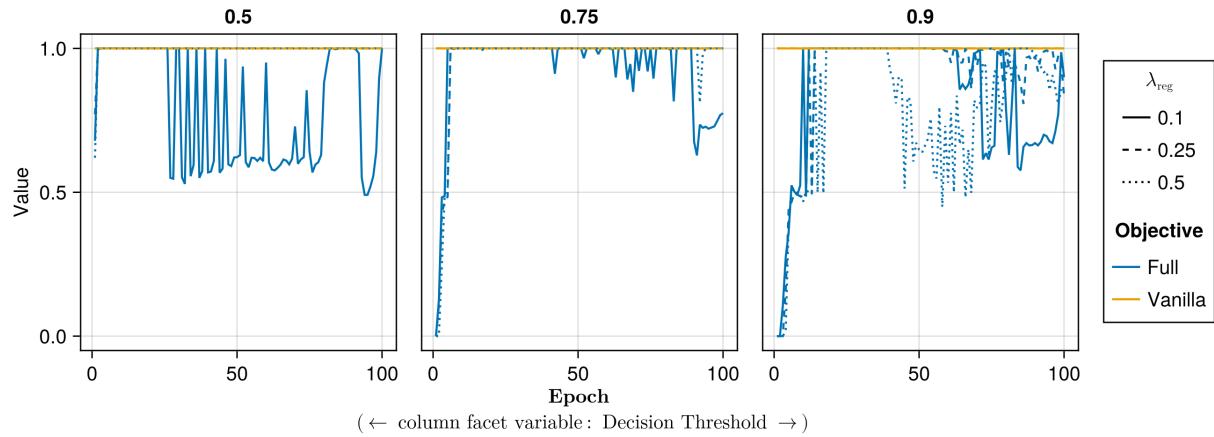


Figure A39: Proportion of mature counterfactuals in each epoch. Data: Circles.

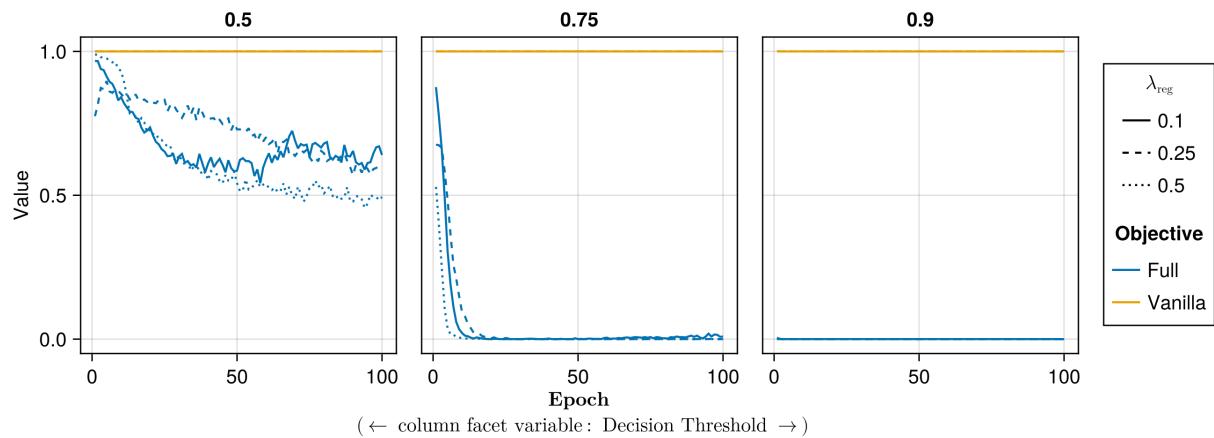


Figure A40: Proportion of mature counterfactuals in each epoch. Data: Credit.

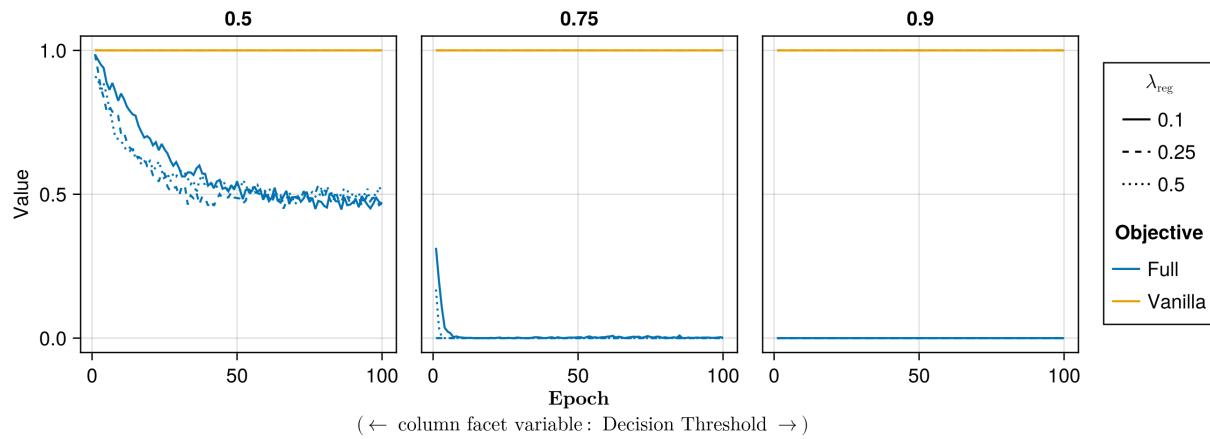


Figure A41: Proportion of mature counterfactuals in each epoch. Data: GMSC.

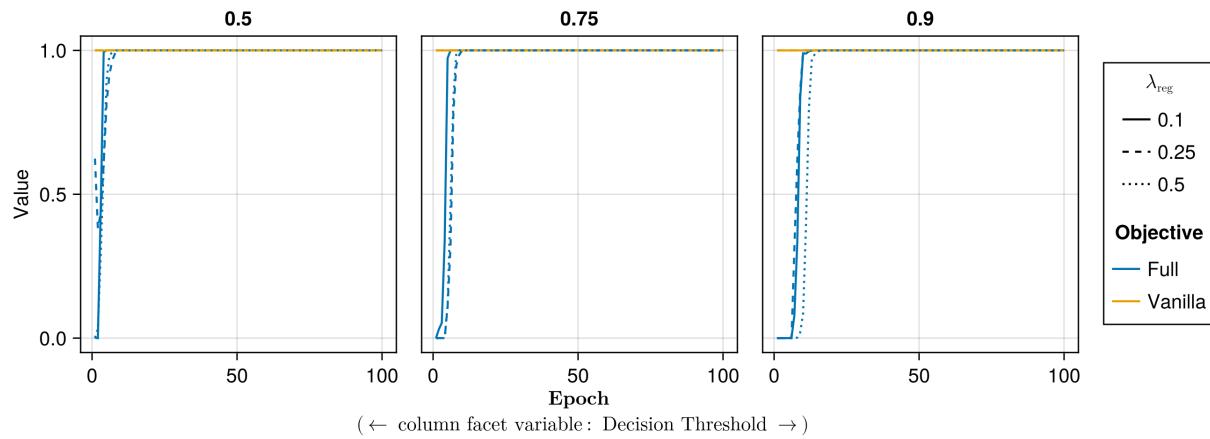


Figure A42: Proportion of mature counterfactuals in each epoch. Data: Linearly Separable.

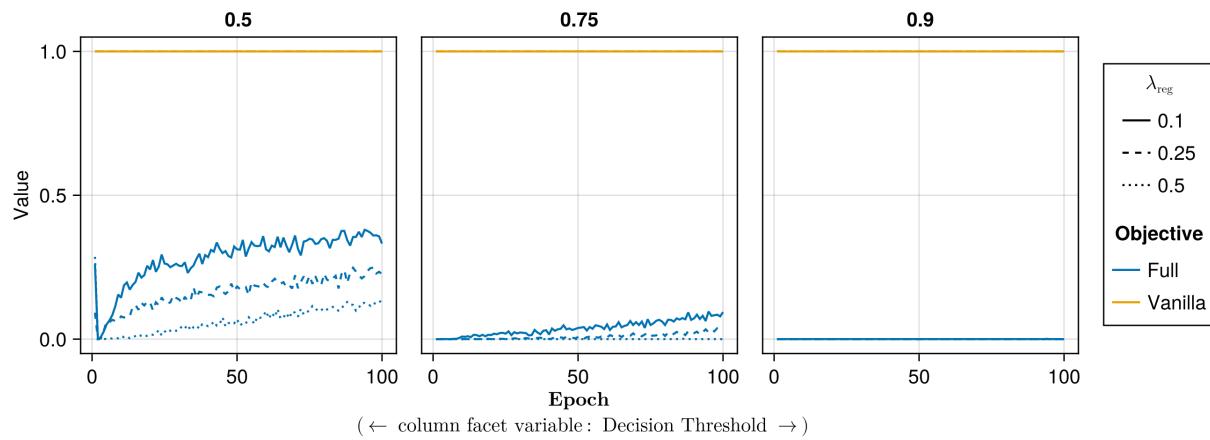


Figure A43: Proportion of mature counterfactuals in each epoch. Data: MNIST.

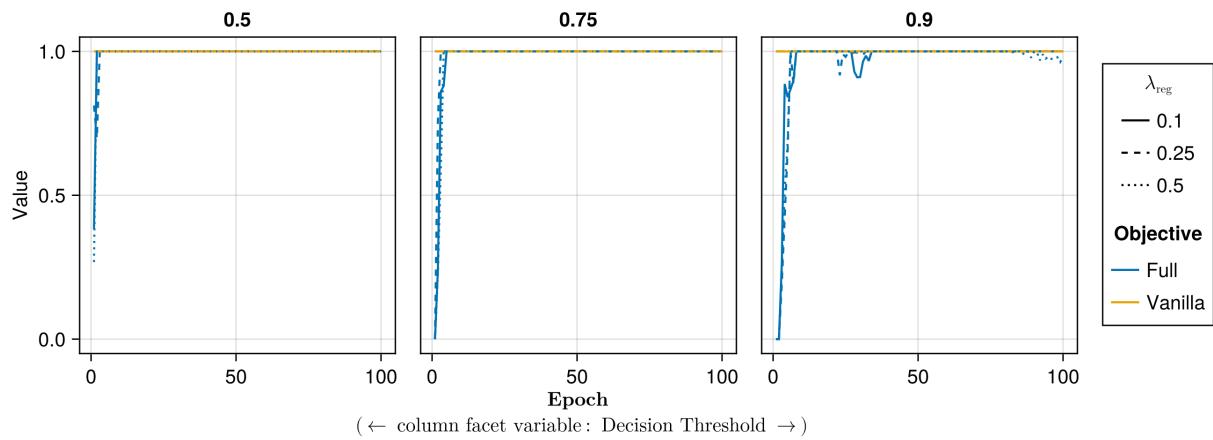


Figure A44: Proportion of mature counterfactuals in each epoch. Data: Moons.

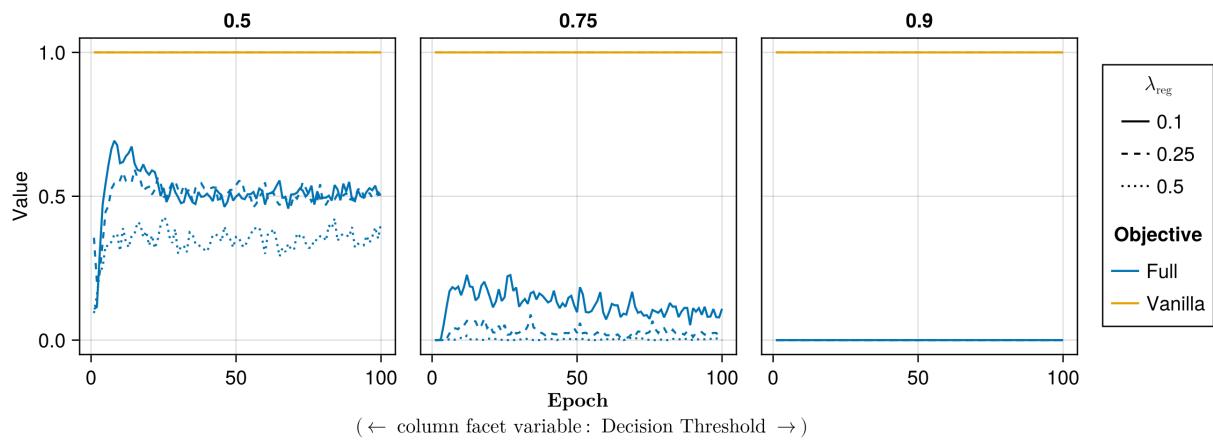


Figure A45: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

Note 9: Training Phase

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

709

Note 10: Evaluation Phase

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

710

711 K.2.1 Plausibility

712 The results with respect to the plausibility measure are shown in Figure A46 to Figure A50.

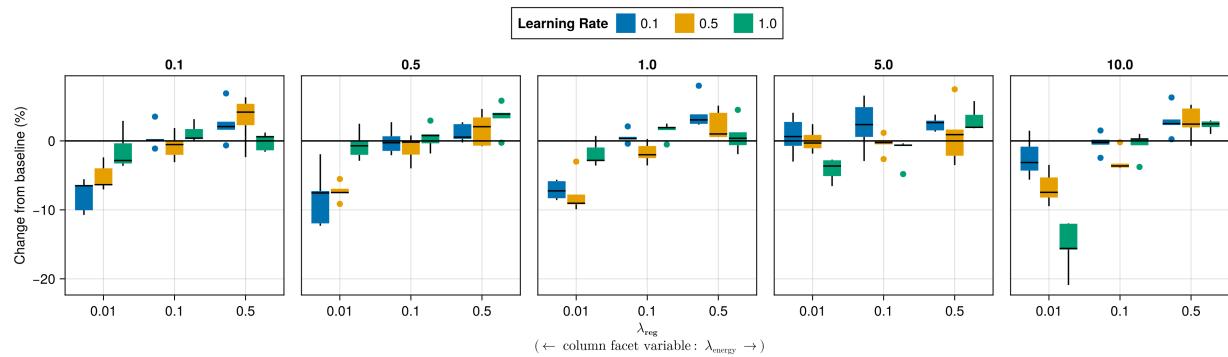


Figure A46: Average outcomes for the plausibility measure across key hyperparameters. Data: Adult.

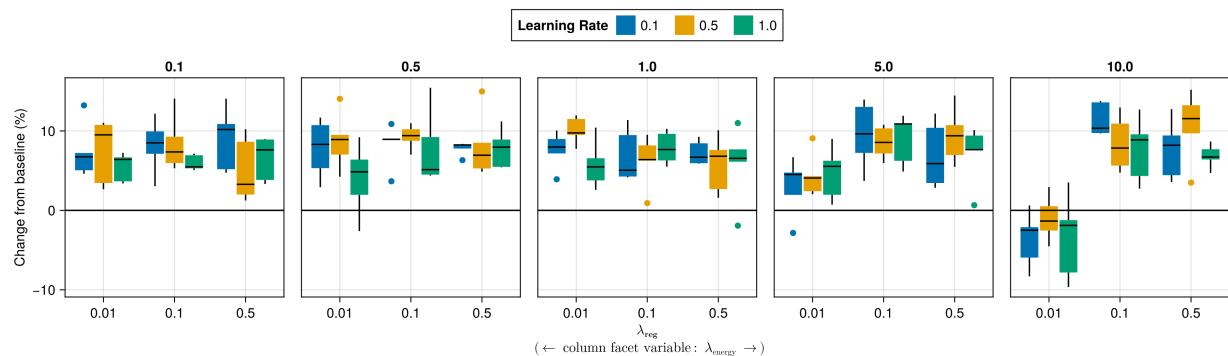


Figure A47: Average outcomes for the plausibility measure across key hyperparameters. Data: Credit.

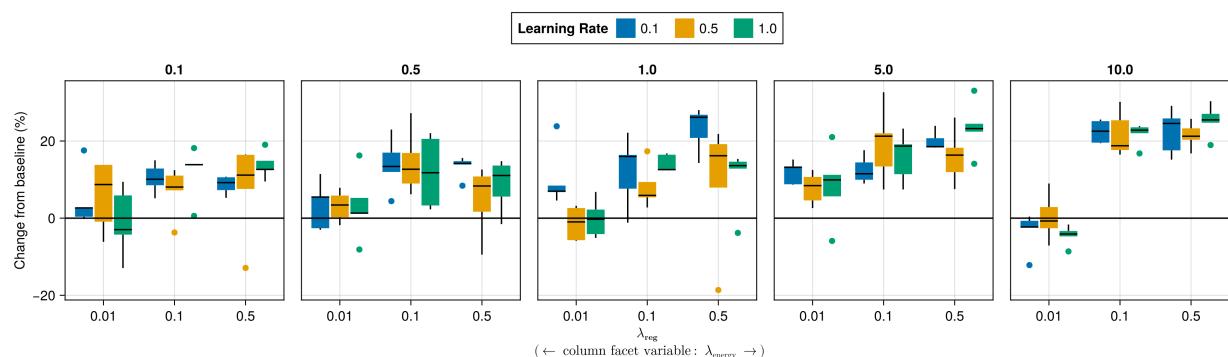


Figure A48: Average outcomes for the plausibility measure across key hyperparameters. Data: GMSC.

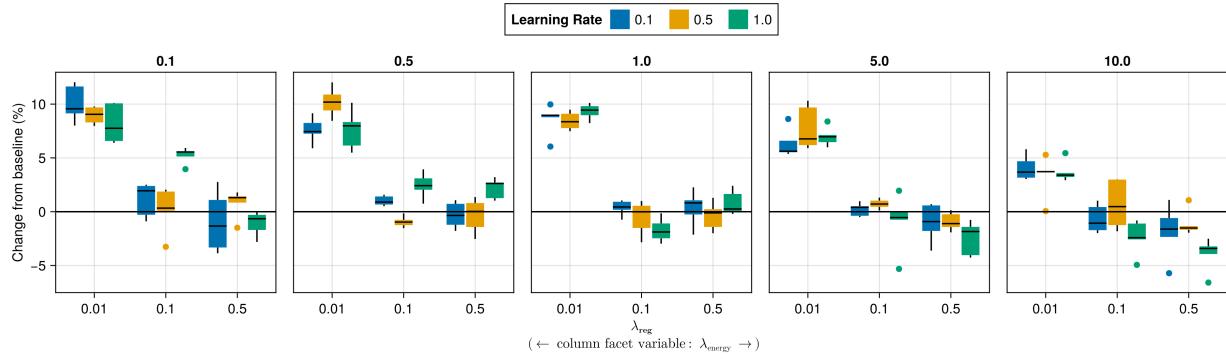


Figure A49: Average outcomes for the plausibility measure across key hyperparameters. Data: MNIST.

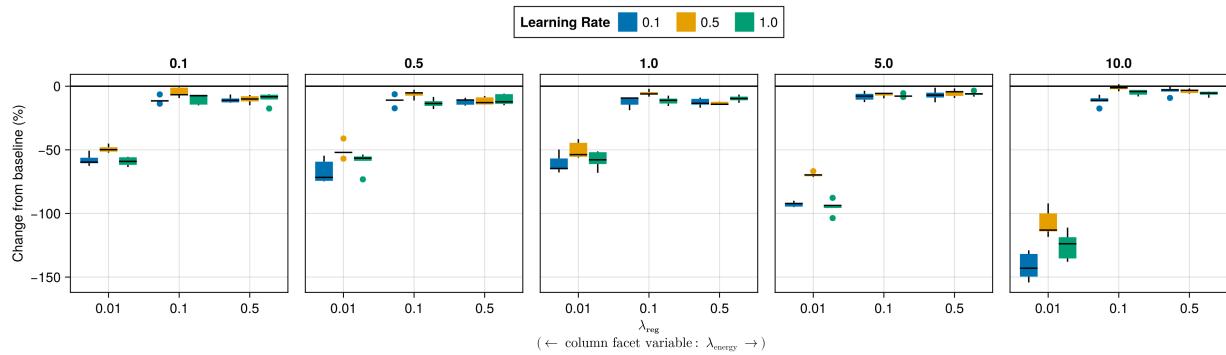


Figure A50: Average outcomes for the plausibility measure across key hyperparameters. Data: Overlapping.

713 K.2.2 Proportion of Mature CE

714 The results with respect to the proportion of mature counterfactuals in each epoch are shown in Figure A51 to Figure A55.
 715

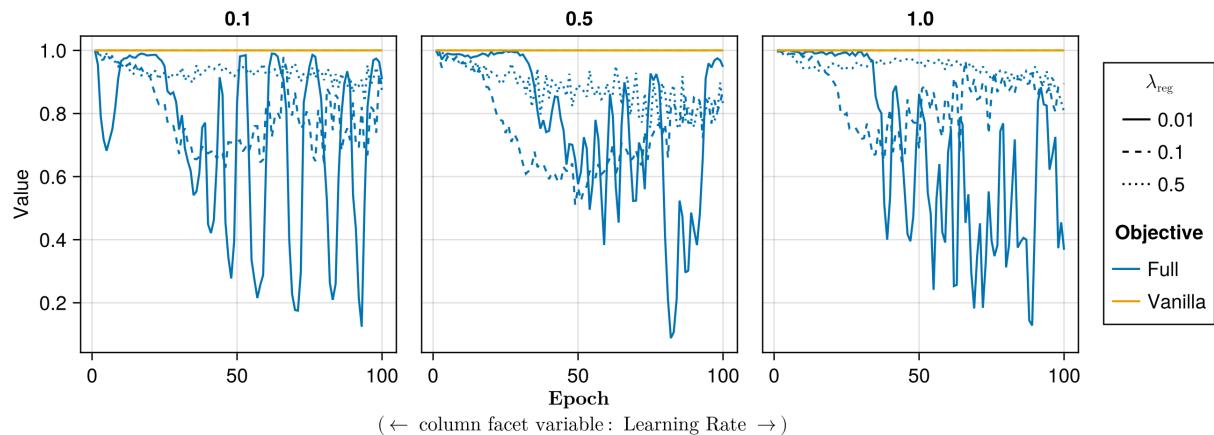


Figure A51: Proportion of mature counterfactuals in each epoch. Data: Adult.

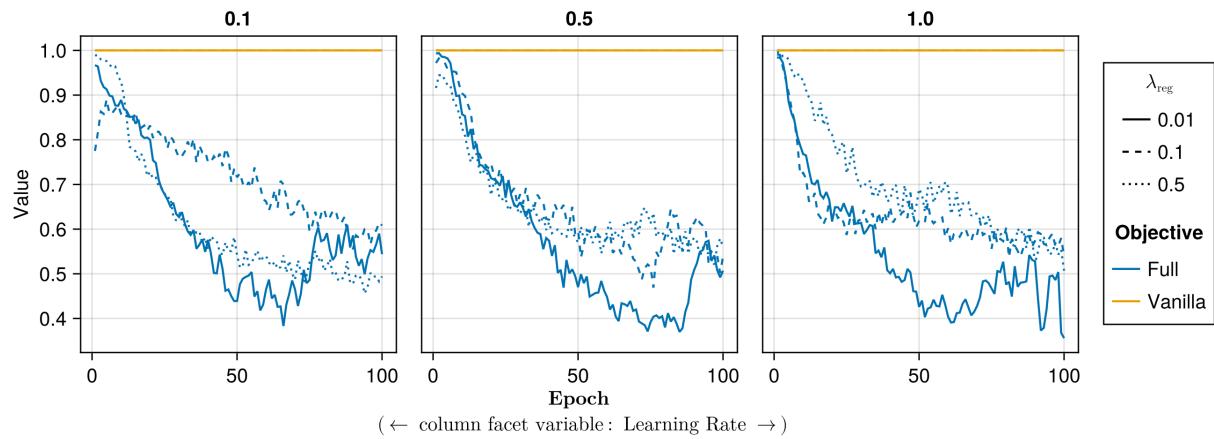


Figure A52: Proportion of mature counterfactuals in each epoch. Data: Credit.

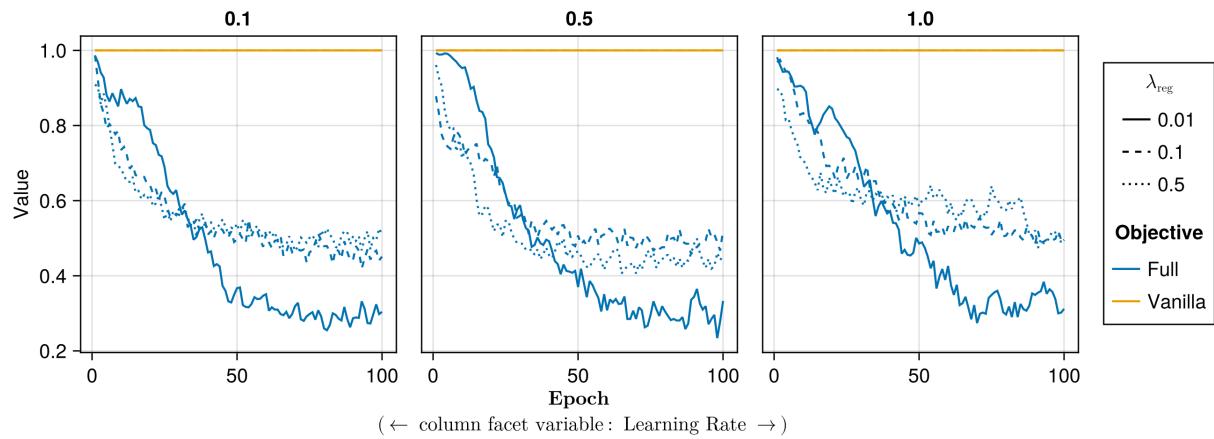


Figure A53: Proportion of mature counterfactuals in each epoch. Data: GMSC.

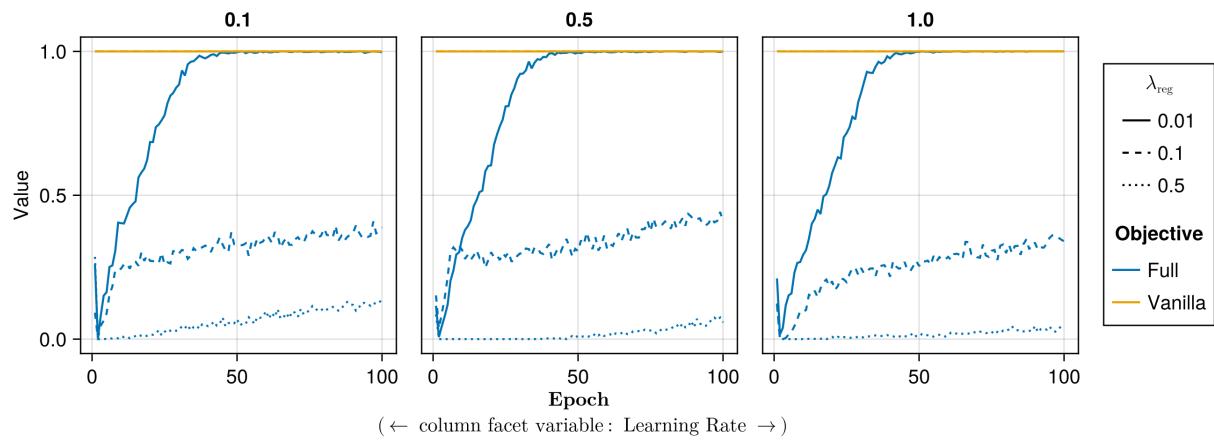


Figure A54: Proportion of mature counterfactuals in each epoch. Data: MNIST.

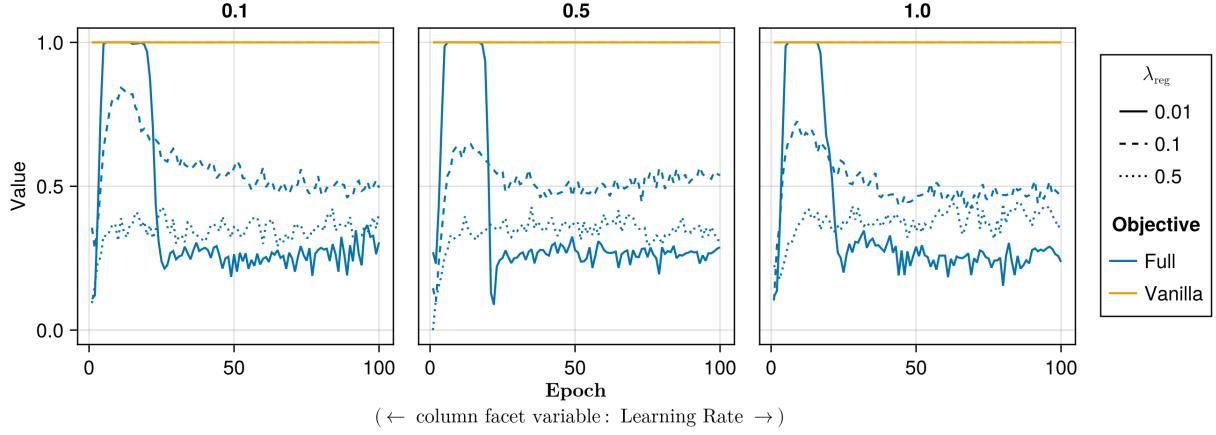


Figure A55: Proportion of mature counterfactuals in each epoch. Data: Overlapping.

716 L Computation Details

717 L.1 Hardware

718 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 719 involve highly parallel tasks and rather small models by today's standard, we have relied on distributed computing 720 across multiple central processing units (CPU). Graphical processing units (GPU) were not required. 721

722 L.1.1 Grid Searches

723 Model training for the largest grid searches with 270 unique parameter combinations was parallelized across 34 CPUs 724 with 2GB memory each. The time to completion varied by dataset for reasons discussed in Section 5: 0h49m (*Moons*), 725 1h4m (*Linearly Separable*), 1h49m (*Circles*), 3h52m (*Overlapping*). Model evaluations for large grid searches were 726 parallelized across 20 CPUs with 3GB memory each. Evaluations for all data sets took less than one hour (<1h) to 727 complete.

728 L.1.2 Tuning

729 For tuning of selected hyperparameters, we distributed the task of generating counterfactuals during training across 40 730 CPUs with 2GB memory each for all tabular datasets. Except for the *Adult* dataset, all training runs were completed 731 in less than half an hour (<0h30m). The *Adult* dataset took around 0h35m to complete. Evaluations across 20 CPUs 732 with 3GB memory each generally took less than 0h30m to complete. For *MNIST*, we relied on 100 CPUs with 2GB 733 memory each. For the *MLP*, training of all models could be completed in 1h30m, while the evaluation across 20 CPUs 734 (6GB memory) took 4h12m. For the *CNN*, training of all models took ~8h, with conventionally trained models taking 735 ~0h15m each and model with CT taking ~0h30m-0h45m each.

736 L.2 Software

737 All computations were performed in the Julia Programming Language (Bezanson et al. 2017). We have developed 738 a package for counterfactual training that leverages and extends the functionality provided by several existing pack- 739 ages, most notably [CounterfactualExplanations.jl](#) (Altmeyer, Deursen, et al. 2023) and the [Flux.jl](#) library for deep 740 learning (Michael Innes et al. 2018; Mike Innes 2018). For data-wrangling and presentation-ready tables we relied on 741 [DataFrames.jl](#) (Bouchet-Valat and Kamiski 2023) and [PrettyTables.jl](#) (Chagas et al. 2024), respectively. For plots and 742 visualizations we used both [Plots.jl](#) (Christ et al. 2023) and [Makie.jl](#) (Danisch and Krumbiegel 2021), in particular 743 [AlgebraOfGraphics.jl](#). To distribute computational tasks across multiple processors, we have relied on [MPI.jl](#) (Byrne, 744 Wilcox, and Churavy 2021).