

---

# COUNTERFACTUAL TRAINING: TEACHING MODELS PLAUSIBLE AND ACTIONABLE EXPLANATIONS

---

A PREPRINT

**Patrick Altmeyer** 

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

[p.altmeyer@tudelft.nl](mailto:p.altmeyer@tudelft.nl)

**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

February 4, 2025

## ABSTRACT

Counterfactual Explanations have emerged as a popular tool to explain predictions made by opaque machine learning models: they explain how factual inputs need to change in order for some fitted model to produce some desired output. Much existing research has focused on identifying explanations that are not only valid but also deemed plausible and desirable with respect to the underlying data and stakeholder requirements. Recent work has shown that under this premise, the task of learning plausible explanations is effectively reassigned from the model itself to the (post-hoc) counterfactual explainer. Building on that work, we propose a novel model objective that leverages counterfactuals during the training phase (ad-hoc) in order to minimize the divergence between learned representations and plausible explanations. Through extensive experiments, we demonstrate that our proposed methodology facilitates training models that inherently deliver plausible explanations while maintaining high predictive performance.

**Keywords** Counterfactual Explanations • 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) models are tasked with learning these representations from scratch, guided by narrow objectives such as predictive accuracy (I. Goodfellow, Bengio, and Courville 2016). Modern advances in computing have made it possible to provide such models with ever greater degrees of freedom to achieve that task, which has often led them to outperform traditionally more parsimonious models. Unfortunately, in doing so they also learn increasingly complex and highly sensitive representations that we can no longer easily interpret.

This trend towards complexity for the sake of performance has come under serious scrutiny in recent years. At the very cusp of the deep learning revolution, Szegedy et al. (2013) showed that artificial neural networks (ANN) are sensitive

23 to adversarial examples: counterfactuals of model inputs that yield vastly different model predictions despite being  
 24 “imperceptible” in that they are semantically indifferent from their factual counterparts. Despite partially effective  
 25 mitigation strategies such as **adversarial training** (I. J. Goodfellow, Shlens, and Szegedy 2014), truly robust deep  
 26 learning (DL) remains unattainable even for models that are considered shallow by today’s standards (Kolter 2023).

27 Part of the problem is that high degrees of freedom provide room for many solutions that are locally optimal with  
 28 respect to narrow objectives (Wilson 2020)<sup>1</sup>. Based purely on predictive performance, these solutions may seem to  
 29 provide compelling explanations for the data, when in fact they are based on purely associative, semantically mean-  
 30 ingless patterns. This poses two related challenges: firstly, it makes these models inherently opaque, since humans  
 31 cannot simply interpret what type of explanation the complex learned representations correspond to; secondly, even  
 32 if we could resolve the first challenge, it is not obvious how to mitigate models from learning representations that  
 33 correspond to meaningless and implausible explanations.

34 The first challenge has attracted an abundance of research on **explainable AI** (XAI) which aims to develop tools to  
 35 derive explanations from complex model representations. This can mitigate a scenario in which we deploy opaque  
 36 models and blindly rely on their predictions. On countless occasions, this scenario has already occurred in practice  
 37 and caused real harm to people who were affected adversely and often unfairly by automated decision-making systems  
 38 (ADMS) involving opaque models (O’Neil 2016). Effective XAI tools can aide us in monitoring models and providing  
 39 recourse to individuals to turn adverse outcomes (e.g. “loan application rejected”) into positive ones (“application  
 40 accepted”). Wachter, Mittelstadt, and Russell (2017) propose **counterfactual explanations** as an effective approach  
 41 to achieve this: they explain how factual inputs need to change in order for some fitted model to produce some desired  
 42 output, typically involving minimal perturbations.

43 To our surprise, the second challenge has not yet attracted any consolidated research effort. Specifically, there has  
 44 been no concerted effort towards improving model **explainability**, which we define here as the degree to which learned  
 45 representations correspond to explanations that are interpretable and deemed **plausible** by humans (see Definition 3.1).  
 46 Instead, the choice has typically been to improve the capacity of XAI tools to identify the subset explanations that are  
 47 both plausible and valid for any given model, independent of whether the learned representations are also compatible  
 48 with implausible explanations (Altmeyer et al. 2024). Fortunately, recent findings indicate that explainability can arise  
 49 as byproduct of regularization techniques aimed at other objectives such as robustness, generalization and generative  
 50 capacity Altmeyer et al. (2024).

51 Building on these findings, we introduce **counterfactual training**: a novel regularization technique geared explicitly  
 52 towards aligning model representations with plausible explanations. Our contributions are as follows:

- 53 • We discuss existing related work on improving models and consolidate it through the lens of counterfactual  
 54 explanations (Section 2).
- 55 • We present our proposed methodological framework that leverages faithful counterfactual explanations during  
 56 the training phase of models to achieve the explainability objective (Section 3).
- 57 • Through extensive experiments we demonstrate the counterfactual training improve model explainability  
 58 while maintaining high predictive performance. We run ablation studies and grid searches to understand  
 59 how the underlying model components and hyperparameters affect outcomes. (Section 4).

60 Despite limitations of our approach discussed in Section 5, we conclude that counterfactual training provides a practi-  
 61 cal framework for researchers and practitioners interested in making opaque models more trustworthy Section 6. We  
 62 also believe that this work serves as an opportunity for XAI researchers to reevaluate the premise of improving XAI  
 63 tools without improving models.

## 64 2 Related Literature

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

---

<sup>1</sup>For clarity: we follow standard ML convention in using “degrees of freedom” to refer to the number of parameters estimated from data.

74 representations that are susceptible to worst-case counterfactuals (i.e. adversarial examples), AT effectively removes  
 75 some implausible explanations from the solution space.

## 76 2.1 Adversarial Examples are Counterfactual Explanations

77 This interpretation of the link between explainability through counterfactuals on one side, and robustness to adversarial  
 78 examples on the other, is backed by empirical evidence. Sauer and Geiger (2021) demonstrate that using counterfactual  
 79 images during classifier training improves model robustness. Similarly, Abbasnejad et al. (2020) argue that counterfactuals  
 80 represent potentially useful training data in machine learning, especially in supervised settings where inputs may  
 81 be reasonably mapped to multiple outputs. They, too, demonstrate the augmenting the training data of image classifi-  
 82 cers can improve generalization. Teney, Abbasnejad, and Hengel (2020) propose an approach using counterfactuals  
 83 in training that does not rely on data augmentation: they argue that counterfactual pairs typically already exist in train-  
 84 ing datasets. Specifically, their approach relies on, firstly, identifying similar input samples with different annotations  
 85 and, secondly, ensuring that the gradient of the classifier aligns with the vector between pairs of counterfactual inputs  
 86 using the cosine distance as a loss function. In the natural language processing (NLP) domain, counterfactuals have  
 87 similarly been used to improve models through data augmentation: Wu et al. (2021), propose *POLYJUICE*, a general-  
 88 purpose counterfactual generator for language models. They demonstrate empirically that augmenting training data  
 89 through *POLYJUICE* counterfactuals improves robustness in a number of NLP tasks. Luu and Inoue (2023) introduce  
 90 Counterfactual Adversarial Training (CAT), which also aims at improving generalization and robustness of language  
 91 models. Specifically, they propose to proceed as follows: firstly, they identify training samples that are subject to  
 92 high predictive uncertainty; secondly, they generate counterfactual explanations for those samples; and, finally, they  
 93 fine-tune the given language model on the augmented dataset that includes the generated counterfactuals.

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

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

## 112 2.2 Beyond Robustness

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

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

129 This is closely related to the idea of aligning models with plausible explanations and has inspired our proposed  
 130 counterfactual training objective, as we explain in Section 3.

### 131 3 Counterfactual Training

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

144 **Definition 3.1** (Model Explainability). Let  $M_\theta : \mathcal{X} \mapsto \mathcal{Y}$  denote a supervised classification model that maps from  
 145 intuts to representations  $\phi(x; \theta)$  and finally to outputs. Assume that for any given input-output pair  $\{x, y\}$  there  
 146 exists a counterfactual  $x' = x + \Delta : M_\theta(x') = y^+ \neq y = M_\theta(x)$  where  $y^+$  denotes some target output. We say that  
 147  $M_\theta$  is **explainable** to the extent that faithfully generated counterfactuals are:

- 148 1. Plausible:  $x' \sim_p \mathcal{X}|y^+$  with  $p \rightarrow 1$  as defined in Altmeyer et al. (2024).  
 149 2. Actionable: permutations  $\Delta$  are subject to actionability constraints.

150 Actionability constraints in Definition 3.1 vary and depend on the context in which  $M_\theta$  is deployed. In this work, we  
 151 focus on domain and mutability constraints for individual features. We also limit ourselves to classification tasks in  
 152 this work, a limitation that we discuss in Section 5.

#### 153 3.1 Our Proposed Objective

154 To train models with high explainability as defined in Definition 3.1, we propose the following objective,

$$\text{yloss}(M_\theta(x), y) + \lambda_{\text{div}} \text{div}(x', x; \theta) + \lambda_{\text{adv}} \text{advloss}(M_\theta(x'), y) \quad (1)$$

155 where  $\text{yloss}(\cdot)$  denotes any conventional classification loss function (e.g. cross-entropy) that induces discriminative  
 156 (predictive) performance. The two additional components in Equation 1 are explained in more detail below. For now  
 157 they can be sufficiently described as inducing explainability directly and indirectly by penalizing: 1) the contrastive  
 158 divergence,  $\text{div}(\cdot)$ , between counterfactuals  $x'$  and observed samples  $x$  and, 2) the adversarial loss,  $\text{advloss}(\cdot)$ , with  
 159 respect to counterfactuals. The tradeoff between the different components can be governed by adjusting the strengths  
 160 of the penalties  $\lambda_{\text{div}}$  and  $\lambda_{\text{adv}}$ .

##### 161 3.1.1 Directly Inducing Explainability through Contrastive Divergence

162 Grathwohl et al. (2020) observe that any classifier can be re-interpreted as a joint energy-based model (JEM) that  
 163 learns to discriminate output classes conditional on inputs and generate inputs. They show that JEMs can perform well  
 164 at both tasks by directly optimizing the joint distribution  $\log p_\theta(x, y) = \log p_\theta(y|x) + \log p_\theta(x)$ . The first factor The  
 165 distribution over inputs is learned via contrastive divergence:

166 Considered through the lens of explainability, JEMs learn plausible explanations.

##### 167 3.1.2 Indirectly Inducing Explainability through Adversarial Robustness

168 A reasonable choice for the latter is to define  $\text{advloss}(M_\theta(x'), y; \varepsilon) := \text{yloss}(M_\theta(x'), y)$

169 **3.2 Encoding Domain Knowledge**

170 **4 Experiments**

171 **4.1 Experimental Setup**

172 **4.2 Experimental Results**

173 **5 Discussion**

174 **6 Conclusion**

175 **References**

- 176 Abbasnejad, Ehsan, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. 2020. “Counterfactual  
177 Vision and Language Learning.” In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition  
178 (CVPR)*, 10041–51. <https://doi.org/10.1109/CVPR42600.2020.01006>.
- 179 Altmeyer, Patrick, Mojtaba Farmanbar, Arie van Deursen, and Cynthia CS Liem. 2024. “Faithful Model Explanations  
180 Through Energy-Constrained Conformal Counterfactuals.” In *Proceedings of the AAAI Conference on Artificial  
181 Intelligence*, 38:10829–37. 10.
- 182 Augustin, Maximilian, Alexander Meinke, and Matthias Hein. 2020. “Adversarial Robustness on in-and Out-  
183 Distribution Improves Explainability.” In *European Conference on Computer Vision*, 228–45. Springer.
- 184 Freiesleben, Timo. 2022. “The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples.”  
185 *Minds and Machines* 32 (1): 77–109.
- 186 Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy. 2014. “Explaining and Harnessing Adversarial Examples.”  
187 <https://arxiv.org/abs/1412.6572>.
- 188 Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- 189 Grathwohl, Will, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swer-  
190 sky. 2020. “Your Classifier Is Secretly an Energy Based Model and You Should Treat It Like One.” In *International  
191 Conference on Learning Representations*.
- 192 Guidotti, Riccardo. 2022. “Counterfactual Explanations and How to Find Them: Literature Review and Benchmark-  
193 ing.” *Data Mining and Knowledge Discovery*, 1–55.
- 194 Guo, Hangzhi, Thanh H. Nguyen, and Amulya Yadav. 2023. “CounterNet: End-to-End Training of Prediction Aware  
195 Counterfactual Explanations.” In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery  
196 and Data Mining*, 577–89. KDD ’23. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3580305.3599290>.
- 197 Kolter, Zico. 2023. “Keynote Addresses: SaTML 2023 .” In *2023 IEEE Conference on Secure and Trustworthy  
199 Machine Learning (SaTML)*, xvi–. Los Alamitos, CA, USA: IEEE Computer Society. [https://doi.org/10.1109/Sa TML54575.2023.00009](https://doi.org/10.1109/Sa<br/>200 TML54575.2023.00009).
- 201 Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. “Simple and Scalable Predictive Uncer-  
202 tainty Estimation Using Deep Ensembles.” *Advances in Neural Information Processing Systems* 30.
- 203 Luu, Hoai Linh, and Naoya Inoue. 2023. “Counterfactual Adversarial Training for Improving Robustness of Pre-  
204 Trained Language Models.” In *Proceedings of the 37th Pacific Asia Conference on Language, Information and  
205 Computation*, 881–88.
- 206 O’Neil, Cathy. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*.  
207 Crown.
- 208 Pawelczyk, Martin, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. 2022. “Exploring  
209 Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis.”  
210 In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, edited by Gustau  
211 Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, 151:4574–94. Proceedings of Machine Learning Research.  
212 PMLR. <https://proceedings.mlr.press/v151/pawelczyk22a.html>.
- 213 Poyiadzi, Rafael, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. “FACE: Feasible and  
214 Actionable Counterfactual Explanations.” In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*,  
215 344–50.
- 216 Ross, Alexis, Himabindu Lakkaraju, and Osbert Bastani. 2024. “Learning Models for Actionable Recourse.” In  
217 *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS ’21. Red  
218 Hook, NY, USA: Curran Associates Inc.
- 218 Sauer, Axel, and Andreas Geiger. 2021. “Counterfactual Generative Networks.” <https://arxiv.org/abs/2101.06046>.
- 219 Schut, Lisa, Oscar Key, Rory Mc Grath, Luca Costabello, Bogdan Sacaleanu, Yarin Gal, et al. 2021. “Generating  
220 Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties.” In  
221 *International Conference on Artificial Intelligence and Statistics*, 1756–64. PMLR.

- 223 Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus.  
224 2013. “Intriguing Properties of Neural Networks.” <https://arxiv.org/abs/1312.6199>.
- 225 Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. “Learning What Makes a Difference from  
226 Counterfactual Examples and Gradient Supervision.” In *Computer Vision–ECCV 2020: 16th European Confer-  
227 ence, Glasgow, UK, August 23–28, 2020, Proceedings, Part x 16*, 580–99. Springer.
- 228 Wachter, Sandra, Brent Mittelstadt, and Chris Russell. 2017. “Counterfactual Explanations Without Opening the Black  
229 Box: Automated Decisions and the GDPR.” *Harv. JL & Tech.* 31: 841. <https://doi.org/10.2139/ssrn.3063289>.
- 230 Wilson, Andrew Gordon. 2020. “The Case for Bayesian Deep Learning.” <https://arxiv.org/abs/2001.10995>.
- 231 Wu, Tongshuang, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. “Polyjuice: Generating Counterfactuals  
232 for Explaining, Evaluating, and Improving Models.” In *Proceedings of the 59th Annual Meeting of the Associa-  
233 tion for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing  
234 (Volume 1: Long Papers)*, edited by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, 6707–23. Online:  
235 Association for Computational Linguistics. <https://doi.org/10.18653/v1/2021.acl-long.523>.

236 **A Training Details**

237 **A.1 Initial Grid Search**

238 For the initial round of experiments we

239 **A.1.1 Generator Parameters**

240 The hyperparameter grids for the first investigation of the effect of generator parameters are shown in Parameters [A.1](#)  
241 and Parameters [A.2](#).

242 **Parameters A.1** (Training Phase).

- 243 • Generator Parameters:
  - 244 –  $\lambda_{\text{cost}}$ : 0.0, 0.001, 0.1
  - 245 –  $\lambda_{\text{div}}$ : 0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0, 15.0
  - 246 – Learning Rate: 1.0
  - 247 – Maximum Iterations: 20, 50, 100
  - 248 – Optimizerimizer: sgd
- 249 • Generator: `ecco`, `generic`, `omni`, `revise`
- 250 • Training Parameters:
  - 251 – Objective: `full`, `vanilla`

252 **Parameters A.2** (Evaluation Phase).

- 253 • Counterfactual Parameters:
  - 254 – Convergence: `max_iter`
  - 255 – Maximum Iterations: 100
  - 256 – No. Individuals: 100
  - 257 – No. Runs: 5
- 258 • Generator Parameters:
  - 259 –  $\lambda_{\text{cost}}$ : 0.0
  - 260 –  $\lambda_{\text{div}}$ : 0.1, 0.5, 1.0, 5.0, 10.0, 20.0
  - 261 – Learning Rate: 1.0
  - 262 – Maximum Iterations: 50
  - 263 – Optimizerimizer: sgd

264 **A.1.1.1 Linearly Separable**

- 265 • **Energy Penalty** (Table [A1](#)): *ECCo* generally does yield better results than *Vanilla* for higher choices of the  
266 energy penalty (10,15) during training. *Generic* performs poorly accross the board. *Omni* seems to have an  
267 anchoring effect, in that it never performs terribly but also never as good as the best *ECCo* results. *REVISE*  
268 performs poorly across the board.
- 269 • **Cost** (Table [A2](#)): Results for all generators (except *Omni*) are quite bad, which can likely be attributed to  
270 extremely bad results for some choices of the **Energy Penalty** (results here are averaged). For *ECCo* and  
271 *Generic*, higher cost values generally lead to worse results.
- 272 • **Maximum Iterations**: No clear patterns recognizable, so it seems that smaller choices are ok.
- 273 • **Validity**: *ECCo* almost always valid except for very low values during training and high values at evaluation  
274 time. *Generic* often has poor validity.
- 275 • **Accuracy**: Seems largely unaffected.

Table A1: Results for Linearly Separable data by energy penalty.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
full	0.01	<i>ECCo</i>	$-9.91 \cdot 10^{11}$	$2.25 \cdot 10^{12}$
full	0.01	<i>Generic</i>	$-5.71 \cdot 10^{17}$	$1.3 \cdot 10^{18}$
<b>full</b>	<b>0.01</b>	<b>Omniscient</b>	<b>-2.54</b>	<b>0.116</b>
full	0.01	<i>REVISE</i>	-15.6	13.2

Continuing table below.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
vanilla	0.01	<i>ECCo</i>	-4.28	3.52
vanilla	0.01	<i>Generic</i>	-4.45	3.47
vanilla	0.01	<i>Omniscient</i>	-5.12	4.46
vanilla	0.01	<i>REVISE</i>	-4.91	4.24
full	0.05	<i>ECCo</i>	$-5.63 \cdot 10^5$	$1.28 \cdot 10^6$
full	0.05	<i>Generic</i>	$-8.35 \cdot 10^{17}$	$1.9 \cdot 10^{18}$
<b>full</b>	<b>0.05</b>	<b>Omniscient</b>	<b>-2.53</b>	<b>0.114</b>
full	0.05	<i>REVISE</i>	-15	12.6
vanilla	0.05	<i>ECCo</i>	-4.4	3.66
vanilla	0.05	<i>Generic</i>	-4.38	3.48
vanilla	0.05	<i>Omniscient</i>	-5.25	4.62
vanilla	0.05	<i>REVISE</i>	-4.94	4.22
full	0.1	<i>ECCo</i>	$-6.74 \cdot 10^5$	$1.53 \cdot 10^6$
full	0.1	<i>Generic</i>	$-1.72 \cdot 10^{11}$	$3.9 \cdot 10^{11}$
<b>full</b>	<b>0.1</b>	<b>Omniscient</b>	<b>-2.56</b>	<b>0.124</b>
full	0.1	<i>REVISE</i>	-15.6	13.2
vanilla	0.1	<i>ECCo</i>	-4.28	3.52
vanilla	0.1	<i>Generic</i>	-4.45	3.48
vanilla	0.1	<i>Omniscient</i>	-5.12	4.46
vanilla	0.1	<i>REVISE</i>	-4.91	4.25
full	0.5	<i>ECCo</i>	-11.8	9.83
full	0.5	<i>Generic</i>	$-1.06 \cdot 10^{18}$	$2.42 \cdot 10^{18}$
<b>full</b>	<b>0.5</b>	<b>Omniscient</b>	<b>-2.54</b>	<b>0.123</b>
full	0.5	<i>REVISE</i>	-15	12.6
vanilla	0.5	<i>ECCo</i>	-4.4	3.65
vanilla	0.5	<i>Generic</i>	-4.38	3.48
vanilla	0.5	<i>Omniscient</i>	-5.25	4.61
vanilla	0.5	<i>REVISE</i>	-4.95	4.22
full	1	<i>ECCo</i>	-11.5	11.1
full	1	<i>Generic</i>	$-1.71 \cdot 10^{11}$	$3.88 \cdot 10^{11}$
<b>full</b>	<b>1</b>	<b>Omniscient</b>	<b>-2.59</b>	<b>0.117</b>
full	1	<i>REVISE</i>	-15.7	13.3
vanilla	1	<i>ECCo</i>	-4.28	3.51
vanilla	1	<i>Generic</i>	-4.44	3.47
vanilla	1	<i>Omniscient</i>	-5.11	4.46
vanilla	1	<i>REVISE</i>	-4.91	4.25
full	5	<i>ECCo</i>	-3.99	3.12
full	5	<i>Generic</i>	$-4.88 \cdot 10^{17}$	$1.11 \cdot 10^{18}$
<b>full</b>	<b>5</b>	<b>Omniscient</b>	<b>-2.53</b>	<b>0.117</b>
full	5	<i>REVISE</i>	-14.6	12.1
vanilla	5	<i>ECCo</i>	-4.4	3.65
vanilla	5	<i>Generic</i>	-4.38	3.48
vanilla	5	<i>Omniscient</i>	-5.25	4.61
vanilla	5	<i>REVISE</i>	-4.95	4.22
<b>full</b>	<b>10</b>	<b>ECCo</b>	<b>-2.31</b>	<b>0.735</b>
full	10	<i>Generic</i>	$-1.7 \cdot 10^{11}$	$3.86 \cdot 10^{11}$
full	10	<i>Omniscient</i>	-2.53	0.117
full	10	<i>REVISE</i>	-15.5	13
vanilla	10	<i>ECCo</i>	-4.28	3.51
vanilla	10	<i>Generic</i>	-4.44	3.47
vanilla	10	<i>Omniscient</i>	-5.12	4.46
vanilla	10	<i>REVISE</i>	-4.91	4.24
<b>full</b>	<b>15</b>	<b>ECCo</b>	<b>-2.01</b>	<b>0.488</b>
full	15	<i>Generic</i>	$-4.91 \cdot 10^{17}$	$1.12 \cdot 10^{18}$
full	15	<i>Omniscient</i>	-2.53	0.116

Continuing table below.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
full	15	<i>REVISE</i>	-14.4	11.7
vanilla	15	<i>ECCo</i>	-4.4	3.65
vanilla	15	<i>Generic</i>	-4.38	3.48
vanilla	15	<i>Omniscient</i>	-5.25	4.6
vanilla	15	<i>REVISE</i>	-4.95	4.23

Table A2: Results for Linearly Separable data by cost penalty.

Objective	$\lambda_{\text{cost}}(\text{train})$	Generator	Value	Std
full	0	<i>ECCo</i>	$-5.32 \cdot 10^3$	$1.21 \cdot 10^4$
full	0	<i>Generic</i>	$-1.03 \cdot 10^{18}$	$2.34 \cdot 10^{18}$
<b>full</b>	<b>0</b>	<b>Omniscient</b>	<b>-2.64</b>	<b>0.125</b>
full	0	<i>REVISE</i>	-15.4	12.9
vanilla	0	<i>ECCo</i>	-4.34	3.58
vanilla	0	<i>Generic</i>	-4.41	3.48
vanilla	0	<i>Omniscient</i>	-5.18	4.54
vanilla	0	<i>REVISE</i>	-4.93	4.23
full	0.001	<i>ECCo</i>	-362	811
full	0.001	<i>Generic</i>	$-2.65 \cdot 10^{17}$	$6.03 \cdot 10^{17}$
<b>full</b>	<b>0.001</b>	<b>Omniscient</b>	<b>-2.49</b>	<b>0.115</b>
full	0.001	<i>REVISE</i>	-15.5	13
vanilla	0.001	<i>ECCo</i>	-4.34	3.58
vanilla	0.001	<i>Generic</i>	-4.41	3.48
vanilla	0.001	<i>Omniscient</i>	-5.18	4.53
vanilla	0.001	<i>REVISE</i>	-4.93	4.23
full	0.1	<i>ECCo</i>	$-3.72 \cdot 10^{11}$	$8.46 \cdot 10^{11}$
full	0.1	<i>Generic</i>	$-4.49 \cdot 10^{14}$	$1.02 \cdot 10^{15}$
<b>full</b>	<b>0.1</b>	<b>Omniscient</b>	<b>-2.5</b>	<b>0.112</b>
full	0.1	<i>REVISE</i>	-14.6	12.2
vanilla	0.1	<i>ECCo</i>	-4.34	3.58
vanilla	0.1	<i>Generic</i>	-4.41	3.48
vanilla	0.1	<i>Omniscient</i>	-5.18	4.54
vanilla	0.1	<i>REVISE</i>	-4.93	4.24

## 276 A.1.1.2 Moons

- **Energy Penalty** (Table A3): *ECCo* consistently yields better results than *Vanilla*, except for very low choices of the energy penalty during training for which it performs abysmal. *Generic* performs quite badly across the board for high enough choices of the energy penalty at evaluation time. *Omni* has small positive effect. *REVISE* performs poorly across the board.
- **Cost (distance penalty)**: *Generic* generally does better for higher values, while *ECCo* does better for lower values.
- **Maximum Iterations**: No clear patterns recognizable, so it seems that smaller choices are ok.
- **Validity**: *ECCo* generally achieves full validity except for very low choices the energy penalty during training and high choices at evaluation time. *Generic* performs poorly for high choices of the energy penalty during evaluation.
- **Accuracy**: Largely unaffected although *ECCo* suffers a bit for very low choices the energy penalty during training. *REVISE* suffers a lot in general (around 10 percentage points).

Table A3: Results for Moons data by energy penalty.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
full	0.01	<i>ECCo</i>	$-2.8 \cdot 10^{22}$	$6.39 \cdot 10^{22}$
full	0.01	<i>Generic</i>	$-4.89 \cdot 10^{30}$	$1.11 \cdot 10^{31}$
<b>full</b>	<b>0.01</b>	<b>Omniscient</b>	<b>-4.74</b>	<b>5.08</b>
full	0.01	<i>REVISE</i>	-572	$1.25 \cdot 10^3$
vanilla	0.01	<i>ECCo</i>	-15.5	17.3
vanilla	0.01	<i>Generic</i>	-10.9	11.9
vanilla	0.01	<i>Omniscient</i>	-12.7	14.4
vanilla	0.01	<i>REVISE</i>	-11.2	13
full	0.05	<i>ECCo</i>	$-1.55 \cdot 10^{16}$	$3.52 \cdot 10^{16}$
full	0.05	<i>Generic</i>	$-2.22 \cdot 10^{20}$	$5 \cdot 10^{20}$
<b>full</b>	<b>0.05</b>	<b>Omniscient</b>	<b>-4.41</b>	<b>4.48</b>
full	0.05	<i>REVISE</i>	$-1.04 \cdot 10^3$	$2.3 \cdot 10^3$
vanilla	0.05	<i>ECCo</i>	-15.5	17.2
vanilla	0.05	<i>Generic</i>	-11.7	12.8
vanilla	0.05	<i>Omniscient</i>	-12.4	14.1
vanilla	0.05	<i>REVISE</i>	-11.3	13.1
full	0.1	<i>ECCo</i>	$-3.41 \cdot 10^3$	$7.73 \cdot 10^3$
full	0.1	<i>Generic</i>	$-5.22 \cdot 10^{30}$	$1.19 \cdot 10^{31}$
<b>full</b>	<b>0.1</b>	<b>Omniscient</b>	<b>-4.78</b>	<b>5.12</b>
full	0.1	<i>REVISE</i>	-288	594
vanilla	0.1	<i>ECCo</i>	-15.5	17.2
vanilla	0.1	<i>Generic</i>	-10.9	11.9
vanilla	0.1	<i>Omniscient</i>	-12.7	14.4
vanilla	0.1	<i>REVISE</i>	-11.3	13.1
full	0.5	<i>ECCo</i>	-7.09	7.51
full	0.5	<i>Generic</i>	$-1.11 \cdot 10^{31}$	$2.53 \cdot 10^{31}$
<b>full</b>	<b>0.5</b>	<b>Omniscient</b>	<b>-4.58</b>	<b>4.83</b>
full	0.5	<i>REVISE</i>	$-1.19 \cdot 10^3$	$2.64 \cdot 10^3$
vanilla	0.5	<i>ECCo</i>	-15.5	17.2
vanilla	0.5	<i>Generic</i>	-11.7	12.8
vanilla	0.5	<i>Omniscient</i>	-12.4	14.1
vanilla	0.5	<i>REVISE</i>	-11.3	13.1
full	1	<i>ECCo</i>	-6.06	6.33
full	1	<i>Generic</i>	$-1.58 \cdot 10^{33}$	$3.59 \cdot 10^{33}$
<b>full</b>	<b>1</b>	<b>Omniscient</b>	<b>-4.66</b>	<b>4.89</b>
full	1	<i>REVISE</i>	$-1.16 \cdot 10^3$	$2.59 \cdot 10^3$
vanilla	1	<i>ECCo</i>	-15.5	17.3
vanilla	1	<i>Generic</i>	-10.9	11.9
vanilla	1	<i>Omniscient</i>	-12.7	14.4
vanilla	1	<i>REVISE</i>	-11.3	13.1
<b>full</b>	<b>5</b>	<b>ECCo</b>	<b>-2.57</b>	<b>2.07</b>
full	5	<i>Generic</i>	$-1.17 \cdot 10^{28}$	$2.66 \cdot 10^{28}$
full	5	<i>Omniscient</i>	-4.29	4.31
full	5	<i>REVISE</i>	-530	$1.16 \cdot 10^3$
vanilla	5	<i>ECCo</i>	-15.5	17.2
vanilla	5	<i>Generic</i>	-11.7	12.7
vanilla	5	<i>Omniscient</i>	-12.4	14.1
vanilla	5	<i>REVISE</i>	-11.3	13.1
<b>full</b>	<b>10</b>	<b>ECCo</b>	<b>-1.76</b>	<b>0.974</b>
full	10	<i>Generic</i>	$-1.54 \cdot 10^{33}$	$3.51 \cdot 10^{33}$
full	10	<i>Omniscient</i>	-4.44	4.56
full	10	<i>REVISE</i>	$-1.52 \cdot 10^3$	$3.4 \cdot 10^3$
vanilla	10	<i>ECCo</i>	-15.5	17.3

Continuing table below.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
vanilla	10	<i>Generic</i>	-10.9	11.9
vanilla	10	<i>Omniscient</i>	-12.7	14.4
vanilla	10	<i>REVISE</i>	-11.3	13.1
<b>full</b>	<b>15</b>	<b>ECCo</b>	<b>-1.37</b>	<b>0.365</b>
full	15	<i>Generic</i>	$-5.32 \cdot 10^{28}$	$1.21 \cdot 10^{29}$
full	15	<i>Omniscient</i>	-4.34	4.38
full	15	<i>REVISE</i>	-473	$1.03 \cdot 10^3$
vanilla	15	<i>ECCo</i>	-15.5	17.2
vanilla	15	<i>Generic</i>	-11.7	12.8
vanilla	15	<i>Omniscient</i>	-12.4	14.1
vanilla	15	<i>REVISE</i>	-11.3	13.1

## 289 A.1.1.3 Circles

- 290
- **Energy Penalty** (Table A4): *ECCo* consistently yields better results than *Vanilla*, though primarily for low to medium choices of the energy penalty ( $<=5$ ) during training. The same goes for *Generic*, which sometimes outperforms *ECCo* (for small energy penalty at evaluation time). *Omni* does alright for lower energy penalty at evaluation time, but loses out for higher choices. *REVISE* performs poorly across the board (except very low choices at evaluation time).
  - **Cost (distance penalty)**: *ECCo* and *Generic* generally achieve the best results when no cost penalty is used during training. Both *Omni* and *REVISE* are largely unaffected.
  - **Maximum Iterations**: *ECCo* consistently yields better results for higher numbers of iterations. *Generic* generally does best for a medium number (50). *Omni* is sometimes invalid (???).
  - **Validity**: *ECCo* tends to outperform its *Vanilla* counterpart, though primarily for low to medium choices of the energy penalty ( $<=5$ ) during training and evaluation. *Vanilla* typically worse across the board.
  - **Accuracy**: Mostly unaffected, but *REVISE* again consistently some deterioration and *ECCo* deteriorates for high choices of energy penalty during training, reflecting other outcomes above.
- 300
- 301
- 302

Table A4: Results for Circles data by energy penalty.

Objective	$\lambda_{\text{div}}(\text{train})$	Generator	Value	Std
<b>full</b>	<b>0.01</b>	<b>ECCo</b>	<b>-1.26</b>	<b>0.423</b>
full	0.01	<i>Generic</i>	-1.49	0.71
full	0.01	<i>Omniscient</i>	-5.21	5.25
full	0.01	<i>REVISE</i>	$-2.71 \cdot 10^{26}$	$6.37 \cdot 10^{26}$
vanilla	0.01	<i>ECCo</i>	-9.33	7.34
vanilla	0.01	<i>Generic</i>	-8.89	6.88
vanilla	0.01	<i>Omniscient</i>	-8.67	6.87
vanilla	0.01	<i>REVISE</i>	-8.65	6.8
full	0.05	<i>ECCo</i>	-1.29	0.397
<b>full</b>	<b>0.05</b>	<b>Generic</b>	<b>-1.21</b>	<b>0.356</b>
full	0.05	<i>Omniscient</i>	-5.08	5.09
full	0.05	<i>REVISE</i>	$-5.91 \cdot 10^{27}$	$1.36 \cdot 10^{28}$
vanilla	0.05	<i>ECCo</i>	-9.35	7.32
vanilla	0.05	<i>Generic</i>	-8.85	6.87
vanilla	0.05	<i>Omniscient</i>	-8.7	6.96
vanilla	0.05	<i>REVISE</i>	-8.52	6.76
<b>full</b>	<b>0.1</b>	<b>ECCo</b>	<b>-1.2</b>	<b>0.383</b>
full	0.1	<i>Generic</i>	-1.5	0.735
full	0.1	<i>Omniscient</i>	-5.17	5.23
full	0.1	<i>REVISE</i>	$-3.06 \cdot 10^{26}$	$7.7 \cdot 10^{26}$
vanilla	0.1	<i>ECCo</i>	-9.33	7.32
vanilla	0.1	<i>Generic</i>	-8.88	6.86
vanilla	0.1	<i>Omniscient</i>	-8.69	6.9

Continuing table below.

<b>Objective</b>	$\lambda_{\text{div}}(\text{train})$	<b>Generator</b>	<b>Value</b>	<b>Std</b>
vanilla	0.1	<i>REVISE</i>	-8.68	6.81
<b>full</b>	<b>0.5</b>	<b>ECCo</b>	<b>-1.12</b>	<b>0.217</b>
full	0.5	<i>Generic</i>	-1.21	0.352
full	0.5	<i>Omniscient</i>	-5.09	5.12
full	0.5	<i>REVISE</i>	$-5.97 \cdot 10^{27}$	$1.37 \cdot 10^{28}$
vanilla	0.5	<i>ECCo</i>	-9.35	7.3
vanilla	0.5	<i>Generic</i>	-8.89	6.92
vanilla	0.5	<i>Omniscient</i>	-8.68	6.93
vanilla	0.5	<i>REVISE</i>	-8.53	6.75
<b>full</b>	<b>1</b>	<b>ECCo</b>	<b>-1.1</b>	<b>0.163</b>
full	1	<i>Generic</i>	-1.49	0.726
full	1	<i>Omniscient</i>	-5.16	5.2
full	1	<i>REVISE</i>	$-3.09 \cdot 10^{26}$	$7.22 \cdot 10^{26}$
vanilla	1	<i>ECCo</i>	-9.34	7.36
vanilla	1	<i>Generic</i>	-8.86	6.85
vanilla	1	<i>Omniscient</i>	-8.7	6.9
vanilla	1	<i>REVISE</i>	-8.69	6.85
full	5	<i>ECCo</i>	-1.75	0.154
<b>full</b>	<b>5</b>	<b>Generic</b>	<b>-1.21</b>	<b>0.363</b>
full	5	<i>Omniscient</i>	-5.14	5.16
full	5	<i>REVISE</i>	$-1.1 \cdot 10^{28}$	$2.5 \cdot 10^{28}$
vanilla	5	<i>ECCo</i>	-9.36	7.32
vanilla	5	<i>Generic</i>	-8.88	6.91
vanilla	5	<i>Omniscient</i>	-8.7	6.93
vanilla	5	<i>REVISE</i>	-8.52	6.73
full	10	<i>ECCo</i>	$-1.02 \cdot 10^6$	$2.32 \cdot 10^6$
<b>full</b>	<b>10</b>	<b>Generic</b>	<b>-1.49</b>	<b>0.702</b>
full	10	<i>Omniscient</i>	-5.13	5.16
full	10	<i>REVISE</i>	$-3.74 \cdot 10^{26}$	$9.09 \cdot 10^{26}$
vanilla	10	<i>ECCo</i>	-9.31	7.33
vanilla	10	<i>Generic</i>	-8.87	6.86
vanilla	10	<i>Omniscient</i>	-8.7	6.89
vanilla	10	<i>REVISE</i>	-8.69	6.83
full	15	<i>ECCo</i>	$-3.31 \cdot 10^{13}$	$7.54 \cdot 10^{13}$
<b>full</b>	<b>15</b>	<b>Generic</b>	<b>-1.22</b>	<b>0.37</b>
full	15	<i>Omniscient</i>	-5.2	5.23
full	15	<i>REVISE</i>	$-9.01 \cdot 10^{27}$	$2.06 \cdot 10^{28}$
vanilla	15	<i>ECCo</i>	-9.38	7.34
vanilla	15	<i>Generic</i>	-8.86	6.87
vanilla	15	<i>Omniscient</i>	-8.69	6.96
vanilla	15	<i>REVISE</i>	-8.51	6.73