Exploring the Connection Between Adversarial Examples and Algorithmic Recourse

Ian Hardy, Alex Salman, Jialu Wang and Xianhang Li
{ihardy,aalsalma,jwang470,xli421}@ucsc.edu

University of California, Santa Cruz

Abstract. Recent work has connected adversarial attack methods and counterfactual explanation methods: both seek minimal changes to an input data point that change a model's classification decision. This raises the question of how often counterfactual explanations are merely producing adversarial examples which "fool" a model, verses examples which represent "faithful" movements towards the desired class. In this work we seek to study this question further by exploring the affect of adversarial training, which discourages a model from being susceptible to adversarial examples, on counterfactual generation using two datasets and two counterfactual explanation methods. We find mixed results that highlight the importance of the counterfactual generation process. The experimental code, as well as the results for this work can be found here

1 Introduction

Despite their remarkable and often super-human performance, many neural networks can be easily "fooled" by introducing small perturbations to inputs that can be invisible to the human visual system. [1] first proposed the concept of "adversarial examples": by adding small perturbations to the input samples, the model obtains incorrect classification results with a high confidence score. These are sometimes referred to as "evasion attacks" [2]. [1] also found that such perturbations can be adapted into different model architectures, further demonstrating that deep neural networks are vulnerable to these input manipulations. Adversarial examples raise concerns about the trust one can place in neural network classifiers, and as such much work has been put into adversarial training methods which discourage the existence of adversarial examples.

There are many methods of adversarial training, the most popular of which [3] generate adversarial examples (with corrected labels) on the fly during training and include them in the model's training set. While adversarial training has been shown to drastically increase robustness to adversarial examples, it often comes at some cost to standard accuracy [4].

As neural networks have been deployed in high-trust environments, another vein of research has appeared which is concerned with finding justified (minimum-cost) means of flipping a model's decision. For example, in the lending setting, if a classifier decides to deny an applicant, the model should provide a feasible set of actions such that that applicant may get approved by undertaking those actions. In the context of accountability, the ability to obtain a desired outcome from a known model, the actionable set of changes the users can make to improve their qualifications, or the systematic process of reversing unfavorable decisions across a range of counterfactual scenarios is defined as "recourse" [5]. These are also often also referred to as "counterfactual explanations."

There is no single method to generate counterfactual explanations as performance highly depends on properties related to the dataset, the model, the application of a score, and factual point specificities [6], recourse should account for the feasibility of different actions that a user could conceivably take. For example, a model should not request actions that violate human rights or dignity (e.g. asking someone to change their martial status from "married" to "single") or that are impossible (e.g. asking someone to decrease their age or education level.) Such restrictions are known as "feasibility constraints." Many "counterfactual explanation" methods do not account for these constraints, or in other ways avoid finding true minimal-cost perturbations.

The goal in this work is to see the effect of adversarial training on counterfactual explanation/ recourse generation. We naturally and adversarially trained models, generated recourse options for both, and measured the success rate, travel distance, and proximity to the desired manifold, of the resulting options.

2 Related work

Adversarial Training and Attack. We first review the basics on adversarial training and two effective attacks. In adversarial machine learning world, our goal is to add invisible perturbation onto the input data, which cannot be detected by humans. The noisy input can readily fool the network to make a wrong decision with a high confidence. Nowadays, we have tons of attack methods in different scenarios. Here we only focus on two wildly-used methods which have been studied for years. First, we have a classifier f with parameters θ. Our input is x and y represents the label we used for supervision. They come from the same distribution D. Our goal is to generate the perturbations δ which can significantly enlarges the cross-entropy loss of $\mathcal{L}$ that typically is used for image classification tasks.

– **FGSM:** [7] first propose Fast Gradient Sign Method (FGSM) to generate the perturbation δ as follows:

$$\delta = \epsilon sign(\nabla_x \mathcal{L}(f_\theta(x), y)), \tag{1}$$

ϵ parameter is controlling the size of noise. Usually it is a very small value. The sign function is operated on the gradient (loss v.s. input), which is used to set the gradient to 1 if it is greater than 0 and -1 if it is less than 0.

– **PGD:** [8] propose a strong iterative version with a random start based on FGSM, name projected gradient descent (PGD) as:

$$x_{t+1} = \mathit{\Pi}_{\|\delta\|_\infty \leq \epsilon} \left(x_t + \alpha \text{sign}(\nabla_{x_t} \mathcal{L}(f_\theta(x_t), y)) \right), \tag{2}$$

where the α denotes the step size of each iteration. PGD provides a better choice for adversarial examples, but it will also cost much more time than FGSM. where ϵ denotes the maximum size of perturbations. In our experiments, we mainly use PGD to generate the adversarial example because of its effectiveness.

Adversarial training has been regarded as one of the most effective strategies to defend against the adversarial threats to machine learning systems. Adversarial Training (AT) was originally proposed by [7] as a method for defending against adversarial attacks. The idea is simple and straightforward: the generated adversarial samples are added to the training set, so that the model learns the adversarial samples combine with the normal examples during the training. [8] first demonstrate the optimization problem in adversarial training and proposes the PGD adversarial attack. Furthermore, there are many advanced adversarial training methods proposed in recent studies. We can formulate the adversarial training as an optimization problem [8] as:

$$\min_\theta \mathbb{E}_{(x,y)\sim D} \big[\max_{\delta \in \mathit{\Delta}} \mathcal{L}(f_\theta(x+\delta), y) \big]. \tag{3}$$

We use this method under PGD attack for our default setting.

Algorithmic Recourse. Our work is closed related to the recourse literature. [9] firstly introduced an integer programming procedure to offer actionable recourse corresponding to a linear classifier. In this context, a decision subject with a feature vector x will seek to flip their prediction outcome $f(x)$ when they are rejected by the classifier f. During this process, the decision subject will make a change $\mathit{\Delta}(x)$ to their profile and incur a cost $\texttt{cost}(x; \mathit{\Delta}(x))$. The learning objective for the decision subject is

$$\begin{aligned} \min \quad & \texttt{cost}(x; \mathit{\Delta}(x)) \\ \textit{s.t.} \quad & f(\mathit{\Delta}(x)) = +1 \\ & \mathit{\Delta}(x) \text{ satisfies feasibility constraints} \end{aligned} \tag{4}$$

A feasible solution of $\mathit{\Delta}(x)$ is called the flip set to the decision subject. Many other solutions have been proposed to solve the recourse objective. [10] proposed to apply a Lagrangian regularization to Equation 4. [11] searches the counterfactual points in the neighborhood of the faithful points under a lower dimensional latent space. [12] proposed to greedily search the N-sphere around the target points until it crosses the decision boundary. [13] critiqued the conceptualization, operationalization and implementation of recourse approaches. [14] discussed the disparity between subpopulations in the recourse process and propose algorithms to mitigate it.

On the relationship between adversarial examples and counterfactual explanations, they are formally described as constrained optimisation problem where the objective is to change the network's output to a some other output by minimally altering the input [15]. Interestingly, in showing relationship between counterfactual explanations and

adversarial attacks, T Freiesleben, [16], argues that the relationship to the true label and the tolerance with respect to proximity are two properties that formally distinguish the two methods. They present a unified mathematical framework that defines commonly used concepts such as flipping/misclassifying and closeness/distance.

Recent work [17] has observed that there is an implicit connection between adversarial examples and counterfactual explanations. Both phenomena seek minimal changes to the input data that "flip" a model's decision, but while adversarial examples are considered undesirable, recourse is generally viewed as positive and, in certain applications, necessary. The key difference is that adversarial examples point to "vulnerabilities" in a model's decision making, while recourse should represent "genuine" shifts from the negative to positive class. There are other differences between the two: counterfactual explanation methods allow for feasibility constraints that limit what features can be changed, while adversarial attacks are generally unconstrained, and while recourse is generally limited to uni-directional classification transitions (from the negative to positive class) adversarial attacks are concerned with transitions between any two classification decisions. Still, this connection raises the question of whether counterfactual explanations are themselves adversarial examples (i.e. whether practitioners are "fooling" their models when generating recourse options) and if so, what methods can be leveraged to improve the faithfulness of counterfactual explanations.

3 Experimental Design & Metrics

The fundamental question we wish to explore in this work is how natural vs adversarial training affects the counterfactual explanations generated from the resulting models. Adversarial examples can be viewed as "unfaithful" counterfactual explanations that do not represent "true" movements towards the desired class, and so our idea is that by adversarially training a model, and thereby discouraging the existence of adversarial examples, we would see more "faithful" counterfactuals generated. Our hypothesis was that adversarial training should lead to counterfactuals which are closer to the desired manifold; at the same time, we recognize that by making it "harder" to flip a classifier's decision in the immediate vicinity of data points, we may be making it harder to generate recourse in the first place. Therefore we also hypothesized that adversarial training would lead to counterfactuals which are further from their original datapoints, and perhaps decrease the overall rate at which they can be identified.

We also want to make note of the difference between counterfactual explanations and recourse. Counterfactual generation can be viewed as an relaxed problem wherein we seek any minimal perturbation that flips the model's classification on a datapoint, whereas recourse is a more strict problem wherein there exists some set of features which cannot be manipulated (e.g. age,) or which can only be manipulated in certain directions (e.g. education level.) To this end, we conducted two sets of experiments: a counterfactual explanation experiment on data with no constraints, and a recourse experiment on data with constraints. Each is described below.

3.1 Relaxed Problem

For the relaxed counterfactual experiment, we explored a binary image classification task using the Fashion-MNIST dataset. We selected two classes (T-Shirts and Trousers) on which to train a binary Neural Network classifier. For the model's architecture, we use a four-layers MLP as backbone. The embedding dimension in the MLP is 64 and expansion ratio is 4. Sepcifically, every two layers consists of a invert bottleneck MLP block ($64 \longrightarrow 256 \longrightarrow 64$). We also equip with the GELU layer between two MLP layers for stable training. For the training details, we use a batchsize of 128, initial learning rate of 0.01 and adopt a cosine learning rate schedule. We train all the models with the SGD for 40 epochs. To adversarially train the model, we used the Projected Gradient Descent (PGD) attack (selected for its speed) with a fairly standard attack radius for this domain (8/255) and for each training batch, adversarially manipulated a random sample of 50% of the instances. Both models achieved very good accuracy (98.9%.) The naturally trained model suffered from susceptibility to attack (43.8% at a radius of 16/255, and 9.5% at a radius of 8/255) while the adversarially was quite resistance (only 3.9% at a radius of 16/255, and 0.7% at a radius of 8/255.) Here we define "susceptibility" as the percentage of test points for which we were able to find an adversarial example within the associated radius.

To generate counterfactuals for the naturally- and adversarially-trained models on the Fashion-MNIST dataset, we selected Growing Spheres (GS) [12] mainly for its speed. We used the CARLA [18] implementation of GS, with default hyperparameters. We also attempted to generate counterfactuals with CARLA's Actionable Recourse (AR) [**?**] and C-CHVAE [19] implementations, but due to the high computational load on this specific domain were unable

to. Even with the efficiency of Growing Spheres, generating counterfactuals on a subsample of 500 data points took over 5 hours.

3.2 Strict Problem

For the strict recourse experiment, we first attempted to train a model on the updated Adult Income dataset [20]; however, without proper feature engineering we found the recourse generated on this problem to be unnatural and uniform. For example, the categorical "occupation code" feature (which had 406 unique values) was a consistent target for the recourse algorithms, which often relied mainly on changing that single feature for flip the model's decision. In the interest of time, and because of our unfamiliarity with the domain of that dataset, we opted to not go through the process of feature engineering on that Adult Income dataset, and instead used the existing Adult Income dataset (which was pre-engineered) within the CARLA package.

We used the default model architecture of the PyTorch NN from the CARLA catalogue (a fully connected series of 3 layers with 13, 9, and 3 hidden units, respectively,) and edited the CARLA package's training capabilities to include an adversarial training procedure similar to the one we described above. Because of the simplicity of the feature set we adversarially trained and tested on a larger radius (16/255.) One interesting wrinkle we considered but did not fully explore is the effect of binary features on adversarial training and recourse. Obviously, due to the attack radius, no binary features could be changed during the adversarial attacks; we are not sure what affect this might have on the model's learning but believe it is an interesting aspect to consider. We considered trying to change the binary features' range to be [0, 16/255] to allow for them to be flipped in the adversarial training procedure, but did not have time. Regardless, both the natural and adversarial models achieved similar accuracy (78.7% and 77.5%, respectively) and while the naturally-trained model experienced a 16.5% susceptibility to attack, the adversarially-trained model only experienced 1.1% susceptibility.

To generate counterfactuals for the naturally- and adversarially-trained models on the Adult Income dataset, we selected Growing Spheres (GS) and Actionable Recourse (AR) based on a linear approximation of the model (AR-LIME) [18]. For GS we used the default hyperparameters, and for AR-LIME the only hyperparameter changed from the default was the flipset size, which we set to 500 in an attempt to achieve the maximum success rate. On Colab GPU resources, it took approximately 8 hours to calculate AR-LIME values on a subsample of 500 data points. Because it look a while to figure out how to edit the CARLA package to support adversarial training, we unfortunately did not have time to run C-CHVAE on the Adult Income dataset as we had initially hoped.

3.3 Metrics

There were a few different metrics we explored to help how adversarial training affects generated counterfactuals or recourse options. They are as follows:

- **Success Rate**: The percentage of points for which a given counterfactual or recourse method was able to identify a point that flipped the model's decision. This metric seeks to estimate the difficulty in identifying a counterfactual explanation or recourse example.
- **Distance Traveled**: The L_2 distance between the original point and the counterfactual generated. This metric too seeks to estimate the difficulty in attaining a counterfactual explanation or recourse example.
- **KNN Manifold Distance**: The average L_2 distance between the generated counterfactual and its top K nearest neighbors of the desired class (i.e. the class of the generated counterfactual.) This metric seeks to measure the proximity of the resulting counterfactuals to the positive class, serving as an estimate of "faithfulness" on the resulting counterfactuals. In our experiments, we use K=5.
- **Sphere Manifold Distance**: The average L_2 distance between the generated counterfactual and its neighbors of the desired class which fell within a given ϵ-ball. This metric too seeks to measure the proximity of the resulting counterfactuals to the positive class, to the same end as the above. In our experiments, the ϵ value is defined as 20% of the average distance from a given point to all other points.
- **Connectedness**: Motivated by [21], we also explore the "connectedness" of the recourse generated for each model. Formally, the authors of [21] define ϵ-connectedness as follows: an instance $e \in X$ is ϵ-connected to an instance $a \in X$ if $f(e) = f(a)$ and if there exists an ϵ-chain $(e_i)_i < N \in X^N$ between e and a such that $\forall n < N, f(e_i) = f(e)$. An ϵ-chain is defined as a finite sequence $e_0, e_1, ...e_N \in X$ between points e and a for a given distance metric d such that $e_0 = e$, $e_N = a$ and $\forall i < N$, $d(e_i, e_i + 1) < \epsilon$.

As a follow-up to our results, we also recorded two pertinent metrics which help reveal the differences between the counterfactual generation methods we explored:

- **Sparsity**: The average number of features changed between the original point and its counterfactual counterpart. Formally for a set of original points $x_1....x_N \in X^D$, and a set of counterfactuals $x'_1....x'_N \in X^D$, we calculate $\frac{1}{N}\sum_{i=1}^{N}\sum_{d=1}^{D}\mathbb{1}(x_i^d \neq x_i^{d\prime})$. This metric is drawn from CARLA.
- **Redundancy**: To calculate redundancy, we flip each feature value in a generated counterfactual x'_i back to its original value in x_i, and see whether the model's decision changes. If it does not, this indicates that the feature was manipulated unnecessarily, and we increment the redundancy counter for x'_i. After calculating the redundancy value for each counterfactual, we average them to get a single value. This metric is also drawn from CARLA.

We also conducted an ablation study on our adversarial training procedure to ensure the appropriateness of our adversarial training procedure. The metrics described above, as well as the results of this ablation study are detailed in the following section.

4 Results

In this section we will simply describe the results of the experiments we ran; in the following section we will discuss them in depth.

4.1 Counterfactuals on Naturally- & Adversarially-Trained Models

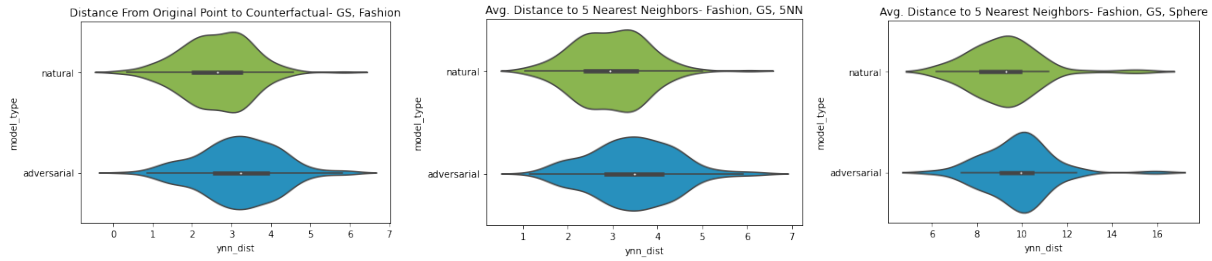
The first metric to cover is the success rate, which is displayed below in Table 1. Next, in Figures 1, 2, and 3 we display the results of running Growing Spheres on the naturally- and adversarially-trained models on the Fashion-MNIST dataset and Growing Spheres and Actionable Recourse on the Adult Income dataset. In each figure, the leftmost subfigure shows the distances distances traveled, the center subfigure shows the KNN manifold distance, and the rightmost subfigure shows the sphere manifold distance. In each subfigure, the upper violin plot displays the results for the naturally-trained model, and the the lower violin plot displays the results for the adversarially-trained model.

In Figure 4 we display the connectedness of the counterfactuals on the naturally- and adversarially-trained models. In the leftmost subfigure we display the connectedness of Growing Spheres on the Fashion-MNIST dataset, in the center subfigure we display the connectedness of Growing Spheres on the Adult Income dataset, and in the rightmost subfigure we display the connectedness of Actionable Recourse on the Adult Income dataset.

Finally, in Table 2 we compare the sparsity and redundancy of Growing Spheres and Actionable Recourse on the Adult dataset. These were calculated on the naturally-trained model; unfortunately, due to a serialization error the results on the adversarially-trained model were overwritten and we did not have time to recalculate them. An immediate follow up work will be to recalculate these and compare them. However for the purpose of discussion they are useful, so we have included them.

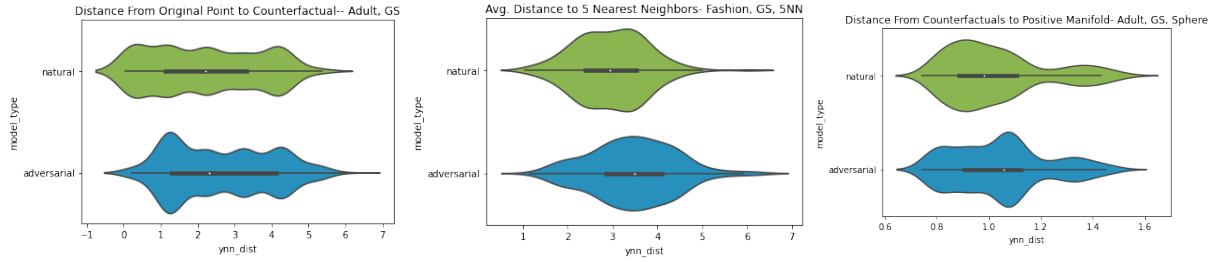
Dataset/ CFE Method	Success Rate (Natural)	Success Rate (Adversarial)
Fashion - Growing Spheres	100%	100%
Adult - Growing Spheres	100%	100%
Adult - Actionable Recourse	37.2%	39.5%

Table 1: **Success Rate**



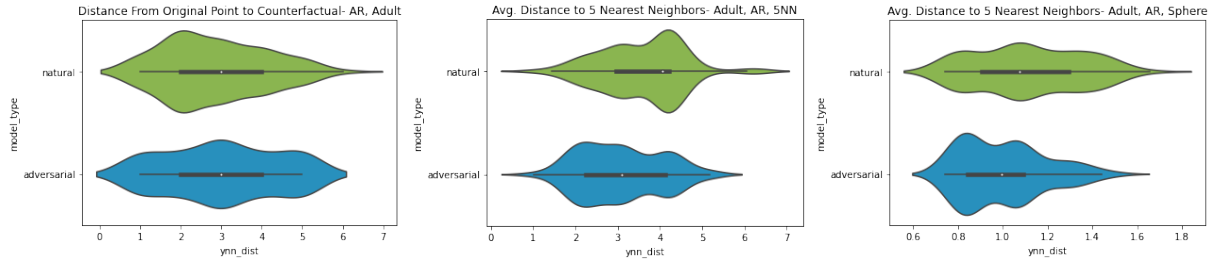
((a)) ((b)) ((c))

Fig. 1: **Growing Spheres on the Fashion-MNIST Dataset**



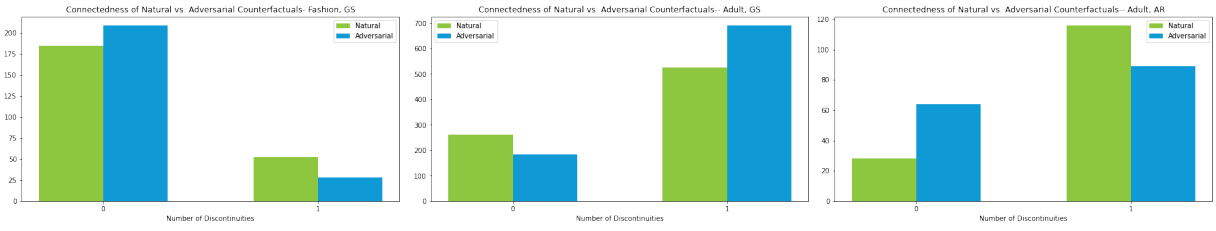
((a)) ((b)) ((c))

Fig. 2: **Growing Spheres on the Adult Income Dataset**



((a)) ((b)) ((c))

Fig. 3: **Actionable Recourse on the Adult Income Dataset**



((a)) ((b)) ((c))

Fig. 4: **Connectedness of Natural vs. Adversarial Counterfactuals**

4.2 Ablation Study

We mainly conduct the ablation study on the Fashion-MNIST Dataset. First, we explore the effect of model size in adversarial attack and adversarial training. Second, we also conduct the experiments on how to generate counterfac-

CFE Method	Sparsity	Redundancy
Growing Spheres	7.09	4.99
Actionable Recourse	1.77	0

Table 2: **Sparsity Redundancy on the Adult Dataset**

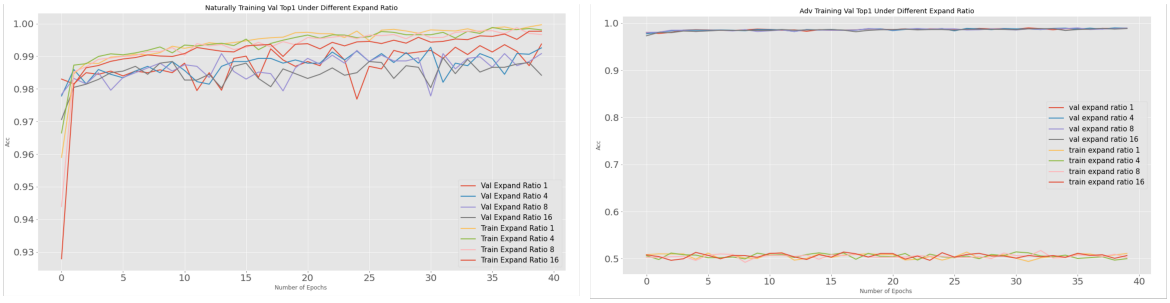


Fig. 5: **Naturally Training v.s Adversarial Training under Different Model Size**.

tuals sample using GS method. Finally, we visualize the adversarial sample and counterfactuals sample to show the difference between the two methods. Through the complete ablation, we aim at demonstrating the full picture of the connection between adversarial and counterfactuals sample under the model perspective.

Adversarial attack and training We first ablate the adversarial training on the model size and hyper-parameters. We choose four sizes of models by adjusting the expansion ratio in the MLP block. Because the scale of dataset is really small, we find the model can over-fit the dataset easily under different settings. As shown in the Fig 5, in the naturally training, the training accuracy can easily reach 100 even using the smallest model. Second, we find the model size can not inluence the clean accuracy much. On the validation set, the Top1 accuracy is close. The similar conclusion can be drew on the adversarial training as well. Here we plot the clean accuracy after adversarial trained, as we mentioned, the over-fitting is servery. Thus, we can clear see the model performance dramatically drop to near 50. The over-fitting problem is common in the adversarial training research. The model can fit on the adversarial examples easily. But this interesting question is out of scope in our project. I have tried several methods like the data augmentation, adding regularization tool. The over-fitting can be alleviated in some extent.

Counterfactuals Generation **Counterfactuals v.s. adversarial samples**. We also visualize the sample generated by different methods. Notice that counterfactuals do not fool a classifier in a classical sense, since individuals need to exert real-world effort to achieve the desired prediction. But in the image classification scenario, we can only change the pixel value to generate a counterfactual example. It is hard to see the real-world effort by alerting a single pixel. Here we would like to show the size of perturbation adding to the original image, which can be useful for us to analysis different mechanisms. As shown in Figure 6, we compare with the natural training sample, adversarial sample and conuterfactual sample. First, it is clear that the counterfactual sample are more noisy than the adversarial sample. We believe the different mechanisms behind two methods are the main reason. The method we used for searching counterfactuals example is growing sphere. We fist generate the random value uniformly on a sphere. In the sphere, we search the point that can successfully generate a counterfactual example. The radius of the sphere is important which can be viewed as the extent of counterfactual. It indicates that the gradient information used in adversarial attack is not used in this searching algorithm. Thus, we can observe much more noise in the counterfactual samples than adversarial example. Because adversarial examples aim to alter the prediction a deep neural network makes on a data point via small and imperceptible changes. The ϵ in the attack is pretty small, which usually is 8/255 that means it can change the 8 pixel values at each step. However, in the counterfactual searching used GS, the step is continuous increasing (the radius is always larger and positive) The second difference is that our adversarial attack is sample-based while counterfactuals are often generated independently of the underlying classification model. It aim to alter data points to suggest impactful changes to individuals. Finally, we compare the natural trained counterfactuals and adversarial trained counterfactuals in the last two columns. We find that the adversarial trained sample usually has larger noise.

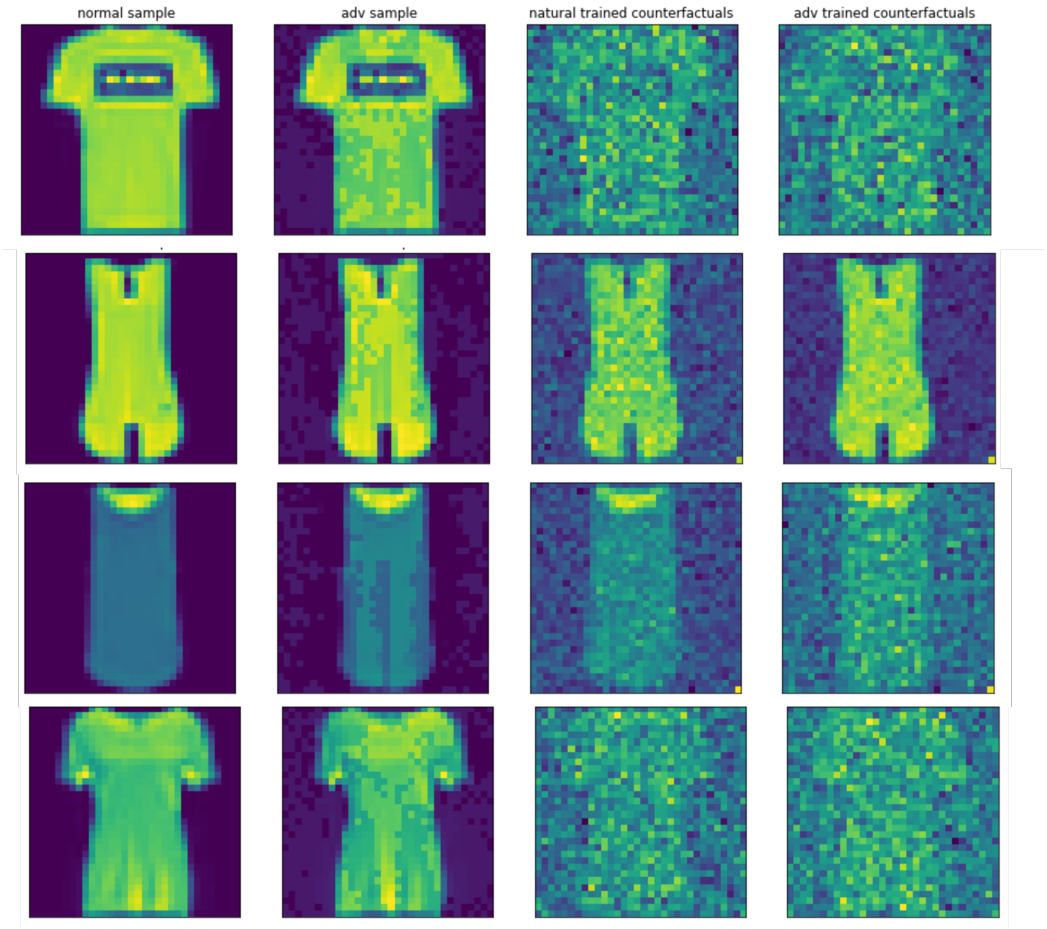


Fig. 6: **Visualization of Different Methods on Fashion-MNIST Dataset**

5 Discussion

In terms of success rate, it seems that adversarial training played little to no role in affecting the ability of the counterfactual generation methods we explored to find counterfactuals that flipped the model's decision. Growing Spheres was able to generate counterfactual explanations for all points we explored, while the rates for Actionable Recourse are quite close.

It is interesting to note, in visualizing the resulting counterfactuals from Growing Spheres on the image set, it appears that the Growing Spheres did little to encourage "faithful" counterfactuals. The counterfactuals generated seem quite noisy, and perhaps can be more accurately described as "adversarial examples" with a large radius than true movements towards the desired class. By this, we mean that no example that we saw seemed to be indicative of a real shift from "T-Shirts" to "Trousers." The affect of adversarial training seems to merely be that the resulting counterfactuals are "noisier" than their natural counterparts. This can perhaps explain the "distance traveled" and "manifold similarity" results for growing spheres on both the Fashion and Adult datasets: in both cases, adversarial training seems to yield counterfactuals that are both further from their original datapoints, as well as further from the desired manifold.

This stands in contrast to the results for Actionable Recourse we found on the Adult dataset. While the resulting counterfactuals tended to travel further from their original datapoints, in the case of both Sphere and KNN distance they are closer they tend to lie closer to the desired manifold. This could perhaps indicate that adversarial training does in fact yield counterfactuals that are more "faithful," at perhaps the cost of more effort to travel from their original values. This result, in turn, raises questions about the ethics and applicability of adversarial training on

models for which recourse is desired. If the recourse generated is more "faithful," but requires more effort on the end-user of the model, when is it appropriate to adversarially train?

The Sparsity and Redundancy values we calculated between Growing Spheres and Actionable Recourse also speak volumes to the difference in the approaches. While Actionable Recourse tended to change between 1 and 2 features on average (in a dataset with 11 actionable features,) Growing Spheres averaged over 7 feature alterations. There were no redundancies found in the Actionable Recourse changes, while Growing Spheres averaged nearly 5 (over 70% of its average change number.) Although it is fast, the random expansion of Growing Spheres means it produces counterfactuals that rarely represent "minimal" changes to flip a model's decision, and raise questions about its applicability in the domain of recourse.

The results of the connectedness studies were a mixed bag. For Growing Spheres on the Fashion dataset, adversarial training cut the number of "unconnected" counterfactuals by about half, while for the Adult dataset it increased them by roughtly 30%. For Actionable Recourse on the Adult dataset, the counterfactuals from the adversarially-trained model experienced roughly 20% fewer "unconnected" points, but in both experiments for the Adult dataset the number of "unconnected" points outweighted the number "connected" points heavily.

6 Conclusion

In this work, we drew on recent work connecting adversarial examples and counterfactual explanations to explore the affect of adversarial training on the counterfactual explanations generated by Growing Spheres (on the Fashion-MNIST and Adult Income datasets) and Actionable Recourse (on the Adult Income dataset.) We find that Growing Spheres produces non-sparse, redundant counterfactuals that seem more like large-radius adversarial examples than minimal-perturbation counterfactual explanations, and that adversarial training seems to lead to counterfactuals which are further from both their original datapoints and the desired manifold. Actionable Recourse, on the other hand, despite its computational expense and lower success rate, produces counterfactuals that are much less sparse and which are not redundant. In the case of Actionable Recourse, we do see some evidence that adversarial training encourages recourse that is closer to the desired data manifold at the cost of more expensive movements to create such examples. A more thorough study is needed to assess the realism of the recourse generated by AR on naturally- and adversarially-trained models, as these results could raise ethical concerns about the cost vs. efficacy of such adjustments.

7 Individual Contributions

Listed below are the individual contributions of our group members:

Written:

- Ian Hardy: Section 3, Section 4.1, Section 5, Section 6
- Alex Salman - Section 1, Section 2
- Jialu Wang - Section 2
- Xianhang Li - Section 4.2

Experimental:

- Ian Hardy: Adversarial Training, Recourse Generation, Metrics
- Alex Salman: Model Training, Metrics
- Jialu Wang: Data Acquisition, Metrics
- Xianhang Li: Model Architecture, Ablation Study

References

1. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.
2. Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018.

3. Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness, 2021.
4. Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy, 2019.
5. Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *ArXiv*, abs/2010.04050, 2020.
6. Raphael Mazzine Barbosa de Oliveira and David Martens. A framework and benchmarking study for counterfactual generating methods on tabular data. *Applied Sciences*, 11(16), 2021.
7. Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.
8. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.
9. Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, jan 2019.
10. Sandra Wachter, Brent Daniel Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Cybersecurity*, 2017.
11. Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. *Proceedings of The Web Conference 2020*, 2020.
12. Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, X. Renard, and Marcin Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *ArXiv*, abs/1712.08443, 2017.
13. Suresh Venkatasubramanian and Mark Alfano. The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 284–293, 2020.
14. Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups. *arXiv preprint arXiv:1909.03166*, 2019.
15. Kieran Browne and Ben Swift. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. *arXiv preprint arXiv:2012.10076*, 2020.
16. Timo Freiesleben. The intriguing relation between counterfactual explanations and adversarial examples. *Minds and Machines*, 32(1):77–109, 2022.
17. Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis, 2021.
18. Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms, 2021.
19. Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, WWW '20, page 3126–3132, New York, NY, USA, 2020. Association for Computing Machinery.
20. Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *arXiv preprint arXiv:2108.04884*, 2021.
21. Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Issues with post-hoc counterfactual explanations: a discussion. *CoRR*, abs/1906.04774, 2019.