The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

ICSE: Interpretable Counterfactual/SemiFactual Explanations

Sooroush Riazi* Mohamad Akbari† Zahed Rahmati‡

Abstract

The lack of explainability and interpretability is one the biggest barriers in real world integration of machine learning. Many researchers tried to improve the interpretability of black box models by post hoc explanations, which try to explain the model by methods like model simplifications, local explanations, explanations by example and many more. Recent user studies showed that Grad-CAM and Lime were less understandable than simple nearest neighbors from the training set. Counterfactual explanations, which are one of the example based explanations have emerged as one of the main methods that can unravel the causal relationship learned in the black box models. to tackle these challenges we propose a novel method to create counterfactual explanations with desired probability in desired class which makes this method more user friendly. In particular, we delve into the concept of semifactual explanations and define near-bound counterfactuals as points with two dominant class probabilities, which makes them closer to the decision boundary. We used Variational AutoEncoders (VAEs) to create latent space and utilize this latent space to find the minimum semantic (not adversarial) change that can change the prediction of instance to any probability in the desired class. However one of the main challenges in creating counterfactuals is the trade-off between the amount of change applied on the instance and the plausibility and interpretability of generated counterfactuals, we conducted experiments on two datasets demonstrating the effectiveness of our approach.

Keywords: Explainable Ai, Counterfactual, and Interpretablity

1 Introduction

Counterfactual explanations have emerged as a prominent trend in the quest for model interpretability. These explanations involve generating instances akin to the original input but with altered feature values, thereby shedding light on the factors that have influence on changing model's output. Recent studies [1] have highlighted the efficacy of explanation-by-example methods across various domains, highlighting their preference over other techniques. The inclination towards explanation-by-example is rooted in human cognition, where examples and counterfactuals play a pivotal role in understanding and planning. Through hypothetical scenarios and what-if analyses, individuals contemplate alternative courses of action and their potential outcomes, a process integral to education, decision-making, and risk assessment.

The formal definition of counterfactual explanations entails making minimal adjustments to an instance to achieve a desired alteration in the outcome. Consider a scenario where a loan request is under scrutiny: for approval, an individual might need to extend their working hours by precisely 10 hours. However, it's imperative to note that altering features solely to provoke a change in prediction, such as changing one's gender, decreasing user age or even indiscriminately increasing working hours without considering other factors like marital status and age, is neither realistic nor aligned with the manifold of our data. Therefore, a methodological approach is necessary to ascertain meaningful alterations that render counterfactuals interpretable. These alterations should not only conform to our training data's manifold but also align with potential variations in the new class manifold, ensuring that counterfactuals remain relevant and insightful within the context of our model's decision-making process.

Another concept sits closely to counterfactuals is semifactuals, Kenny (2020) pointed out the importance of semi factual explanations and lack of research in this area. An example of semi-factual explanations is "Even if you had double your current salary, your loan would still have been refused". Kenny (2020) argued that these explanations capture important causal information regarding the prediction as they bring insight to the decision boundary of the classifier and help users interact with the models and make more informative decisions.

They also presented a method called Plausible Exceptionality-based Contrastive Explanations (PIECE) which automatically models the distributions of latent features to detect "exceptional features", features that play an important role in an instance's prediction, modifying them to be "normal" in explanation

*Department of Mathematics and Computer Science, Amirkabir University of Technology, sooroush.riazi@gmail.com
† Department of Mathematics and Computer Science, Amirkabir University of Technology, akbari.mo@aut.ac.ir
‡ Department of Mathematics and Computer Science, Amirkabir University of Technology, zrahmati@aut.ac.ir

generation. The method finds semi-factuals by finding the last step before a change happens in the model's prediction, but sets no constraint on getting predictions with two dominant classes, initial class and desired class, that's what we call near bound examples, examples that are closest to get 50-50 prediction in two classes. We can also use these examples to improve our model with samples near the decision boundaries where classification is really hard.

To be able to create meaningful Counterfactual and Semifactuals, we need to create a latent space that holds semantics and characteristics of data rather than applying the changes in the feature space. When we are moving around in latent space and trying to create semantic change on an instance we ideally assume we can find a direction which can increase the characteristics of desired class but when our latent space is entangled there is a high possibility that the created changes increase the characteristics of other classes as well, entanglement can be a **big challenge** in finding optimized counterfactuals in latent space, new methods try to address this challenge by different approaches, we mixed triplet loss with VAE to create disentangled latent spaces and analyze our approach in different settings.

In this problem we often have a trade off between these two objectives:

1. Creating a counterfactual with high probability in desired class .

2. Minimizing the amount of change we want to apply on the instance.

As Arnaud and Klaise (2019) mentioned " Often a trade off needs to be made between sparsity and interpretability of CF", we understood the amount of change we need to apply on an instance to create counterfactual is highly dependent to the instance it self so we cannot find the perfect weights for these two terms so we can balance them for all the instances we are trying to explain.

In response to these challenges (The Trade Off and the entanglment), our approach employs Variational AutoEncoders (VAEs) to construct a latent space that captures the underlying semantics of the data. We generate counterfactuals by optimizing a two step search loss that can fix the trade off mentioned above or using curved interpolation that address the entanglement challenge to find the sample with minimum loss in interpolated lines in the latent space. We then address the challenge of entanglement of latent space and compare the counterfactuals generated in each space.

Finally, to validate our method we compare it with the SOT approaches and use counterfactuals in the domain of model debugging, where we introduce human centered procedures to annotate generated hard samples (near bound counterfactuals) and improve classifiers by training them on hard samples.

The main contributions of this paper are threefold

- Creating Smooth transformation from instance to counterfactual which will enable the user to understand the exact result of small changes on prediction that will lead to counterfactual and let him/her choose the exact amount of change to overcome the challenge caused by entanglement in Latent space of VAE, we introduce two solutions: curved interpolations and disentangling the latent space using triplet loss vae.

- Introducing a novel optimization method for creating interpretable counterfactuals with desired probabilities and classes with a two step solution that addresses the trade off between distance and changing the prediction.

- Introducing the concept of near bound counterfactuals, and utilizing a loss function to create CFs with only initial class and desired class dominant probabilities, focusing on new applications for semi factual generation methods we introduce human centered iterative procedures to annotate generated hard samples (near bound counterfactuals) and improve classifiers by training them on hard samples.

2 Related Work

Post hoc explanations in machine learning, including techniques like SHAP, LIME, and feature attribution methods, have garnered considerable attention for their ability to shed light on model predictions. Recent studies emphasize the effectiveness of example-based explanations, highlighting their superiority over other methods. Counterfactual generation, rooted in the work of Wachter et al., initiated the trend towards explanation-by-example methodologies. However, early approaches faced limitations, prompting advancements such as Van Looveren and Klaise's prototype-guided counterfactuals. These innovations aimed to improve interpretability by integrating plausibility criteria into the explanation generation process.While existing research predominantly focuses on counterfactual explanations, Kenny introduced the concept of semifactuals, highlighting their potential significance in understanding model predictions. Despite their promise, a precise methodology for generating semifactuals remained elusive. Notably, Saeed Khorram's Cycle-Consistent Counterfactuals by Latent Transformations introduced a nonlinear framework for generating counterfactuals, enriching the landscape of explanation methodologies.

Among the existing techniques, Van Looveren and Klaise's approach stands out for its utilization of prototypes to guide the counterfactual generation process. By leveraging prototypes, their method

addressed the interpretability and plausibility challenges inherent in early counterfactual approaches.

However, the lack of precise methods for semi factual explanations underscored a gap in the literature, which Kenny highlighted with the PIECE framework.

Saeed Khorram's Cycle-Consistent Counterfactuals by Latent Transformations introduced a novel paradigm, emphasizing nonlinear transformations to generate counterfactuals. This approach diverges from traditional linear methods, offering a fresh perspective on explanation generation.

In response to these developments, our work builds upon the foundational concepts of counterfactual and semifactual explanations. We propose a comprehensive framework that integrates disentangled latent spaces, near-bound counterfactuals, and human-centered model debugging procedures to enhance the interpretability and trustworthiness of machine learning models. By leveraging advancements in disentanglement learning and near-bound counterfactuals, we aim to address the limitations of existing methods while advancing the frontier of interpretable AI.

In summary, the evolution of counterfactual and semifactual explanations reflects a concerted effort to unravel the decision-making processes of machine learning models. Each advancement, from prototype-guided approaches to nonlinear transformations, contributes to the ongoing quest for transparent and trustworthy AI systems.

3 Preliminaries

Before diving into details we need to define several basic concepts in regards to latent space and generation models.

3.1 Entanglement

(Na et al., 2023), entanglement denotes mixing or interdependence of latent factors with respect to labels within a model's space, such as a Variational Autoencoder (VAE). In an entangled representation, the data points of all classes are mixed up in the latent space, hindering to find a direction that will only increase a desired class membership. Mitigating entanglement is a key objective in contrastive learning, as it aims to foster the independence and interpretability of latent factors, enhancing the model's ability to capture meaningful and separate distribution of classes in various latent spaces.

3.2 Prototypes

Building upon the methodology proposed by Van Looveren & Klesse (2020), prototypes are defined for a given instance x_0. Initially, the predictive model is invoked to label the dataset with the classes predicted by the model. Subsequently, for each class i, the instances belonging to that class are encoded and ordered based on their increasing ℓ_2 distance to $ENC(x_0)$. Analogous to the approach outlined in (Snell et al., 2017), the class prototype is determined as the average encoding over the K nearest instances in the latent space sharing the same class label.

4 Methodology

We defined the counterfactual generation as an optimization problem where we want to minimize the amount change applied to an image while shifting the class of that image to the desired class, to mathematically model this problem we incorporated two diffrent approach one models the problem geometric and the other is a two step optimization process.

1. curved interpolations
2. latent space optimization

4.1 Curved interpolation:

When we have a semantic latent space for our data a line can mean a semantic transformation from one point to another, the power of interpolations in augmenting new samples has been demonstrated by methods like SMOTE, Chawla et al. (2002).

But first we must understand why simple interpolation can not be useful in counterfactual generation problem, when we have a simple line from an instance to destination in desired class we may encounter many other classes in the line in other words the semantic change of this line may not only be in favor of our destination class, this problem is also referred to entanglement of latent space which was defined earlier. so we need our line to be able to dodge other classes to have a clean transformation.

We model counterfactual problem as finding a line with curves from given instance to a destination which is inside our desired class distribution in latent space, additionally we do want any point of this line to be decoded as either initial class of instance or destination class.

We first define the process of how we find our destination for given instances LCR and then the Curve interpolation process.

4.1.1 Latent Class Representative (LCR)

We will define a new representative for each class for any given instance, we define LCR like prototypes but with one more condition that each of the K nearest neighbors of each class we choose must be reconstructed and then classified in the desired class by our predictive model. In formal terms, we define, for each unique

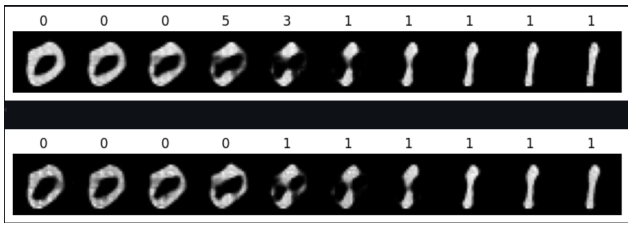

Figure 1: First row: Normal interpolation in latent space Second row: Curved interpolation method in latent space

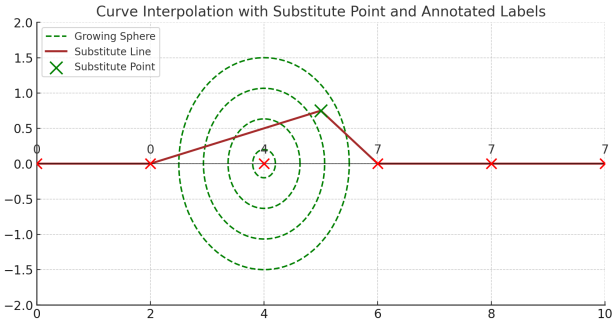

Figure 2: Curve Interpolation Process

class i, instances belonging to that class are encoded and arranged based on their increasing ℓ_2 distance to $ENC(x_0)$. Each Latent Class i Representative is defined as the average encoding over the K nearest instances in the latent space whose reconstructions yield a class i prediction by our predictor model which we are trying to explain. By ensuring that our class prototypes are accurately reconstructed within the correct class, we can ascertain the discovery of a perfectly interpretable counterfactual. This counterfactual may not necessarily involve minimal changes for the input instance across all classes.

$$\text{LCR}_i := \frac{1}{K}\sum_{k=1}^{K}\text{ENC}(x_{ik})$$

$$\text{where} \quad \text{PRED}(\text{REC}(\text{ENC}(x_{ik}))) = i$$

4.1.2 Algorithm for Curve Interpolation Process

1. Identify the Local Closest Region (LCR) for the given instance to serve as the destination.
2. Create a straight line from the instance to the LCR.
3. Divide the line into N arbitrary points.
4. Detect the first entanglement occurrence:
   - An entanglement occurrence is defined as a point along the line that, when decoded into feature space, is predicted as a different class from the initial or destination classes.
5. Search for a substitute for the first entanglement occurrence:
   (a) Generate M random points within a small-radius sphere centered around the entanglement point.
   (b) Check if any of the generated points are classified as either the initial or destination class after decoding.
   (c) If such points exist, select the one closest to the previous point along the main line to replace the entanglement point.
   (d) If no valid points are found, increase the radius incrementally and repeat step 5(a).
6. Return the set of points forming the final interpolated curve.

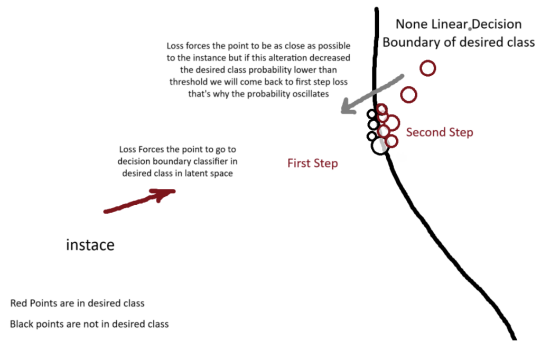

Figure 3: Demonstration of Counterfactual Latent Space Membership Search

4.2 Counterfactual Latent Space Membership Search

This method is a two step optimization, the first step is only about changing the prediction, we focus solely on decreasing the cross entropy loss with the desired class as a gold label to the given threshold. then we start decreasing the distance of the last step image to a starting instance while not changing its predicted probabilities. As this two step solution free us from all the parameters that controlled the trade off between minimizing applied change and interpretability of resulting CF, we now can create the best counterfactual distance wise while holding desired interpretability. Finally By introducing near bound counterfactuals and near bound search loss, we utilize the same optimization process to create counterfactual and semi factual explanations with two dominant probabilities. This way we can focus on the notion of semi factuality directly.

4.2.1 Interpretable Counterfactual Loss Function

To identify and generate counterfactuals within the desired class, we introduce a loss function that guarantees both the alteration of prediction and minimal perturbations. We utilize cross-entropy and reconstruction error to ensure the interpretability of counterfactuals. When our counterfactual exhibits a high probability in the desired class, we regard it as more interpretable. The problem has two parameters which have very non linear relation with each other so it was very difficult to find a suitable parameter for the optimization problem, as there is a trade off between distance and interpretability.

Even for different instances we would need different parameters as some instances are closer to the destination they should get to. To solve this problem without the use of adaptive parameters we propose a two step optimization solution. We understood that changing the prediction must happen first since it is the main task, after changing the class of input instances we can focus on minimizing the distance of the created point and starting instance while locking the cross entropy loss in desired value.

If loss_ce $>$ Th, then final_loss $= 100 \cdot$ loss_ce; otherwise,

$$\text{final\_loss} = D_l + 8 \cdot D_f + 10 \cdot L_{ce} + 5 \cdot \text{global\_rec\_err} + 20 \cdot \text{cr\_p}$$

- L_{ce} is the cross-entropy loss of the generated point with the desired class as the gold label.
- Th is the threshold that we want for our prediction in the desired class. This threshold decides between changing the prediction and minimizing the distance.
- D_l is the distance in latent space, defined as the ℓ_2 norm of the distance between the instance and the current point.
- D_f is the ℓ_2 norm distance in feature space.
- `global_rec_err` is the reconstruction error of the current point using the Global VAE.
- `cr_p` is the Class VAE reconstruction penalty, defined as:

$$\text{penalty} = \text{RELU}(\text{rec\_err}_{t0\_class\_vae} - \text{rec\_err}_{ti\_class\_vae})^2$$

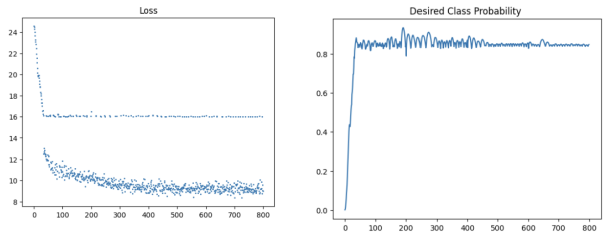

Figure 4: Loss and Desired Class Probability while using Inter- pretable Counterfactual method

4.2.2 Near Bound Counterfactual Search Loss:

In defining counterfactuals, we previously focused solely on minimizing changes that induce a change in prediction, without direct consideration of their proximity to the decision boundary. By utilizing cross-entropy in the loss, we aimed to create counterfactuals with minimal change that exhibit a single dominant prediction. However, to address the issue of proximity to the decision boundary alongside minimal changes, we aim to minimize the difference between initial and desired class probabilities in classifier predictions while also minimizing the sum of probabilities of all other classes. To enforce the change in counterfactual class predictions, we include $(p_d - p_i)$ in the loss function. By adopting these adjustments, we can create counterfactuals with two dominant predictions.

$$\text{final\_loss} = 15 \cdot (pd - pi)^2 + 20 \cdot (\text{sum})^2 + 5 \cdot (pd - pi)$$

$$+12 \cdot \left(\frac{2}{3} \cdot \text{distance} + \frac{1}{3} \cdot \text{distance\_l} \right)^2 + 5 \cdot \text{rec\_err}$$

4.2.3 The Optimization Process

We used the Adam optimization to move in the latent space of our encoder and optimize our loss function. It was argued that moving in latent space can cause semantic changes and this desirable property lets us achieve interpretable counterfactual by guidance of gradients of the classifier model toward desired class boundary in latent space. We will multiply cross entropy loss by a big number like 1000 to make sure the distance values don't take all the attention of the optimizer.

This will cause discontinuities in our loss values seen in figure 3. When we minimize the distance we will usually increase initial class probability and we may pass the threshold of cross entropy but if this happens the optimizer will find the fastest way to get back to the desired threshold. It is expected that the probability of desired class will start by a low level and increase to get to threshold and then **oscillate** around the threshold while minimizing the distance during this optimization. We divided the step size by 2 each 200 epoch to find fine grain changes that minimizes the distance.

5 Experiments

We have designed the following experimental study in order to answer the following research questions:

- RQ1: Can ICSE optimization method outperform state-of-the-art counterfactual methods based on the former quantitative counterfactual metrics?
- RQ2: How much and in what way different disentanglement methods can affect ICSE and other SOT CF generation methods ?
- RQ3: Can Human in the loop Near Bound Counterfactual Generation pipeline help us debug and improve our classification models ?

5.1 Setup MNIST and Fashion-MNIST.

We evaluate the ICSE method against CF explanation baselines, namely, Contrastive Explanation Method (CEM), Counterfactual Visual Explanation (CVE) on the MNIST and Fashion-MNIST datasets by both qualitative inspection and an extensive set of quantitative metrics. Images from both datasets have 28×28 resolutions in 10 classes. We use the standard train/test split. use the examples from the query and CF classes in the train set ($\sim$ 6,000 samples/class) for training and the examples from the query/CF class in the test set ($\sim$ 1,000 samples/class) for evaluation. We used official implementation of our method which were avialble in the github, our source code for experiments is available at `https://github.com/sooroushr/Counterfactual_Explainations` for other researchers to reproduce the results.

5.2 Inspection of the Counterfactuals

In this section, we list the evaluation metrics and their interpretations in the realm of counterfactuals.

- **L1 & L2 Distance**: The L1 norm distance of the instance and the generated counterfactual. This distance helps us find the amount of change applied to the instance.
- **Reconstruction Error with General VAE**: This metric calculates the general plausibility of generated counterfactuals. We train a VAE on the whole training set and use it to calculate the reconstruction error. We find the L2 norm of the distance between the counterfactual and its reconstruction using the VAE and report it as the reconstruction error of the counterfactual.
- **Latent Space Distance**: The L1 and L2 norm distances between the instance and the generated

counterfactual in the latent space of the general VAE. This distance demonstrates similarities in characteristics between the instance and the counterfactual.

- **IM1**: Let VAE_{ti} and VAE_{t0} be Variational Autoencoders trained solely on the instances of class ti and t0 from the training data, where t0 is the initial class of the instance we are trying to find a counterfactual for and ti is the destination class that the generated counterfactual must belong to. We consider a counterfactual interpretable if it belongs to class ti much more than class t0. One way to calculate this is with the prediction model, but single class trained VAE's also help us calculate how much the characteristics of the given image fit the semantics of the class it belongs to. Formally, IM1 measures the ratio between the reconstruction errors of x_{cf} using VAE_{ti} and VAE_{t0}.

  A lower value for IM1 means that x_{cf} can be better reconstructed by the autoencoder which has only seen instances of the counterfactual class i than by the autoencoder trained on the original class t0. This implies that x_{cf} lies closer to the data manifold of counterfactual class i compared to t0, which is considered to be more interpretable.

$$\text{IM1}(\text{AE}_i, \text{AE}_{t0}, x_{cf}) := \frac{\|x_0 + \delta - \text{AE}_i(x_0 + \delta)\|_2^2}{\|x_0 + \delta - \text{AE}_{t0}(x_0 + \delta)\|_2^2 + \epsilon}$$

- **Human Acceptance:** if a human considers the change and the generated counterfactual valid and a real number in desired class.
- **Time**: The amount of time it took the method to create the counterfactual.

5.3 Research Question 1 (RQ1)

RQ1: Can the ICSE optimization method outperform state-of-the-art counterfactual methods?

To answer this question, we designed a fair experiment to compare the counterfactuals generated by ICSE to two other methods: CEM and CF Proto. The CEM method implemented in the Alibi Python library does not accept a destination class, so we first used this method in our experiment to get a counterfactual and then used the class of the generated counterfactual as the destination class for ICSE and CF Proto. We evaluated all of the quantitative metrics for 42 samples for each MNIST dataset (digits and fashion) and method.

5.3.1 Distance

the comparative analysis revealed CEM applies really small amount of change on the instance, this change

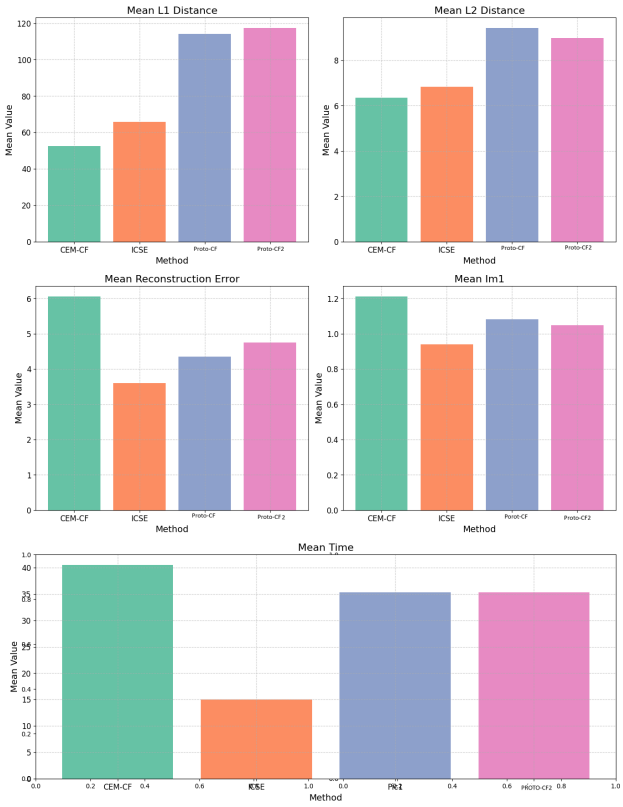

Figure 5: Mean Evaluation Metrics for 42 Test samples with Each Method

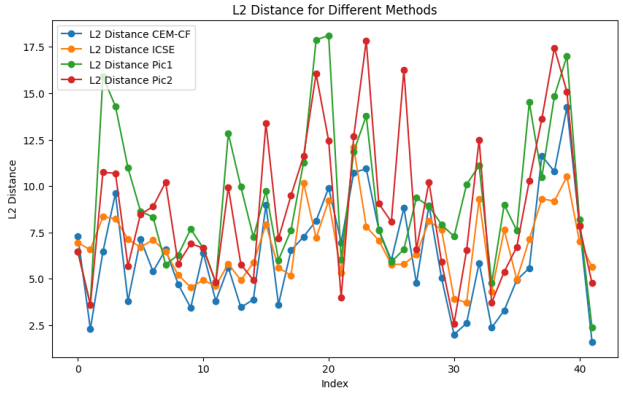

Figure 6: L2 Distance of Different Methods on 42 images from test set

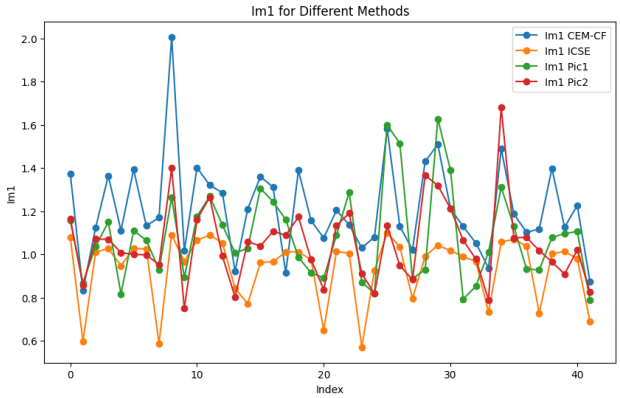

Figure 7: IM1 Score of different Methods on 42 images from test set

is usually strong and does not seems like a noise, but sometimes the change is a complete noise and it seems like the method failed.

ICSE is goes tow to tow with CEM in distance metrics alike CEM the Counterfactual generated by ICSE are semantically changed and the change does not seems like noise at all also in all of 42 instances only 2 of ICSE counterfactuals were human unacceptable while CEM had 20 human unacceptaple counterfactuals. we can see that CEM has worst metrics in means of IM1 and Reconstruction Error.

5.3.2 Interpretablity

ICSE method has lowest IM1 Mean which shows that this method is the most interpretable method amongst the others, the important point is ICSE has lower Distance and IM1 compared to other methods and this means that it is much more testable and the changes are the most small and intertptable possible changes.

Figure 8: Instance and CF generated by CF Prototype Method

Figure 9: instance and CF generated by CEM

Figure 10: instance and CF generated By ICSE

5.4 Research Question 2 (RQ2)

RQ2: What is the effect of disentanglement in the ICSE explanations ?

disentangling the latent space could be a solution to the entanglement challenge.

5.4.1 Triplet Loss VAE

we implemented triplet loss vae to control the entanglement of the generated latent space. in this method we add triplet loss to the loss of VAE, triplet loss gets an anchor and positive and negative sample and decrease the distance of anchor and positive sample while increasing the distance of anchor and the negative sample this will lead to same class data points to create clusters in the latent space.

$$\mathcal{L}_{\text{triplet}} = \max(0, \|f(x_a) - f(x_p)\|_2^2 - \|f(x_a) - f(x_n)\|_2^2 + \alpha)$$

where:

- x_a is the anchor sample,
- x_p is the positive sample,
- x_n is the negative sample,
- $f(\cdot)$ is the embedding function (encoder in the VAE),
- α is the margin.

5.4.2 Total Loss for Triplet Loss VAE

The total loss for the Triplet Loss VAE is a combination of the VAE loss and the triplet loss.

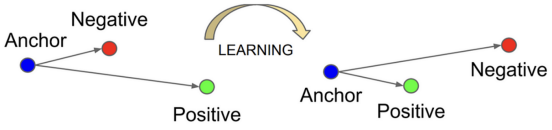

Figure 11: Illustration of Triplet loss

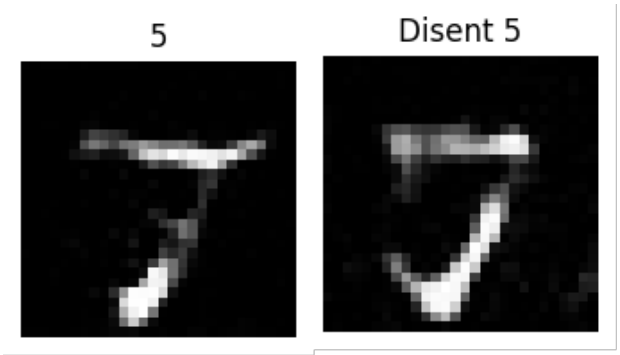

Figure 12: An example of cf generated in entangled and dis- entangled space using optimization method

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{VAE}} + \lambda \mathcal{L}_{\text{triplet}}$$

where λ is a weighting factor to balance the VAE loss and the triplet loss.

5.4.3 Comparative Analysis

Our hypothesis was by employing Triplet Loss VAE we can create a disentangled latent space and by changing the margin of the triplet loss we can create different amount of disentanglement. another way to measure the disentanglement is to train a classifier on the embedding of data and evaluate the classification results of that classfier. We tested on different spaces and compared the metrics of cf to find out the effect of entanglement in this process:

Table 1: Average Metrics for Margin 3, 150 test samples

Metric	Entangled Mean	Disentangled Mean
l1_distance	**88.558487**	96.287669
l2_distance	8.782893	**8.612775**
rec_err	**3.912788**	4.625215
IM1_Mean	1.014836	**0.920788**

based on the human analysis we found the margin 3 disentangled latent space better than vanila latent space thus showcasing that disentangling the latent space would create more interpretable counterfactuals but too much disentanglement may remove the sections of latent space which is a joint space between classes (entanglement) and removing these ambiguous counterfactuals and increase the interpretablity but this is with the cost of increasing the amount of change applied.

Table 2: Average Metrics for Margin 20, 150 test samples

Metric	Entangled Mean	Disentangled Mean
l1_distance	**83.520761**	107.862631
l2_distance	**8.334036**	9.232445
rec_err	**4.062162**	4.641649
IM1_Mean	0.962065	**0.925619**

5.5 Research Question 3 (RQ3)

RQ3: Can we use Near Bound Counterfactuals in a human-in-the-loop process for debugging and improving classification models?

For this experiment, we first imbalanced the MNIST dataset by removing 90% of the data for one of the classes (class 9). We aim to investigate the power of augmenting our data using the near-bound counterfactual (NB CF) method. These counterfactuals are designed to be close to the decision boundary between two classes and are hard samples that the model is more likely to misclassify.

By labeling the misclassified NB CFs using a human-in-the-loop process, we aim to debug the model and improve its accuracy, especially on the imbalanced class. Labeling these near-boundary points helps the model find the decision boundary more efficiently and with fewer data points. To test this hypothesis, we augmented 60 samples using three methods: SMOTE [**?**], NB CF method from ICSE, and using real data.

We trained the same model on all datasets for 10 epochs and then compared the metrics to see which method improved the model better.

Table 3: Class 9 Classification Report on different datasets

Dataset	Class 9 Precision	Class 9 Recall	Class 9 F1
Imbalanced	100	92	95
Smote	99	95	97
NB CF	100	**94**	**97**
Real Data	100	95	97

6 Discussion

We have demonstrated the effectiveness of our method for data augmentation and using the information and characteristics of two classes to create a new point in desired class, most of data augmentation methods in imbalanced datasets only use the minor samples for augmentation but using the counterfactual methods we can

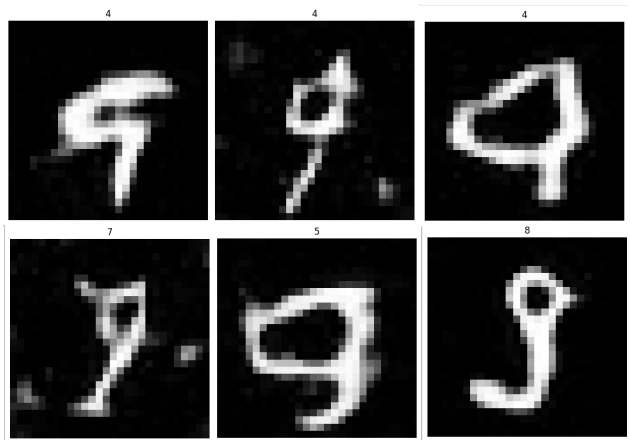

Figure 13: Examples of misclassified NB counterfactuals

create a new data augmentation technic which would have more power than current methods.

References

[1] Arnaud, V. L., & Klaise, J. (2019, July 3). *Interpretable counterfactual explanations guided by prototypes*. arXiv.org. https://arxiv.org/abs/1907.02584

[2] Dhurandhar, A., Chen, P., Luss, R., Tu, C., Ting, P., Shanmugam, K., Das, P. (2018, February 21). Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. arXiv.org. https://arxiv.org/abs/1802.07623

[3] Ribeiro, M. T., Singh, S., Guestrin, C. (2016, February 16). "Why should I trust you?": explaining the predictions of any classifier. arXiv.org. https://arxiv.org/abs/1602.04938

[4] Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[5] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. International Journal of Computer Vision, 128(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

[6] Khorram, S., Fuxin, L. (2022, March 28). Cycle-Consistent counterfactuals by latent transformations. arXiv.org. https://arxiv.org/abs/2203.15064

[7] Hoffer, E., Ailon, N. (2014, December 20). Deep metric learning using Triplet network. arXiv.org. https://arxiv.org/abs/1412.6622