

Augmentation by Counterfactual Explanation - Fixing an Overconfident Classifier

Sumedha Singla*
 University of Pittsburgh
 sumedha.singla@pitt.edu

Sofia Triantafyllou
 University of Crete
 sof.triantafyllou@gmail.com

Nihal Murali*
 University of Pittsburgh
 nihal.murali@pitt.edu

Forough Arabshahi
 Meta AI
 forough@meta.com

Kayhan Batmanghelich
 University of Pittsburgh
 kayhan@pitt.edu

Abstract

A highly accurate but overconfident model is ill-suited for deployment in critical applications such as healthcare and autonomous driving. The classification outcome should reflect a high uncertainty on ambiguous in-distribution samples that lie close to the decision boundary. The model should also refrain from making overconfident decisions on samples that lie far outside its training distribution, far-out-of-distribution (far-OOD), or on unseen samples from novel classes that lie near its training distribution (near-OOD). This paper proposes an application of counterfactual explanations in fixing an over-confident classifier. Specifically, we propose to fine-tune a given pre-trained classifier using augmentations from a counterfactual explainer (ACE) to fix its uncertainty characteristics while retaining its predictive performance. We perform extensive experiments with detecting far-OOD, near-OOD, and ambiguous samples. Our empirical results show that the revised model have improved uncertainty measures, and its performance is competitive to the state-of-the-art methods.

1. Introduction

Deep neural networks (DNN) are increasingly being used in *decision-making* pipelines for real-world high-stake applications such as medical diagnostics [6] and autonomous driving [7]. For optimal decision making, the DNN should produce accurate predictions as well as quantify uncertainty over its predictions [8, 38]. While substantial efforts are made to engineer highly accurate architectures [23], many existing state-of-the-art DNNs do not capture the uncertainty correctly [9].

We consider two types of uncertainty: *epistemic uncer-*

tainty, caused due to limited data and knowledge of the model, and *aleatoric uncertainty*, caused by inherent noise or ambiguity in the data [30]. We evaluate these uncertainties with respect to three test distributions (see Fig 1):

- **Ambiguous in-Distribution (AiD):** These are the samples within the training distribution that have an inherent ambiguity in their class labels. Such ambiguity represents high aleatoric uncertainty arising from class overlap or noise [60], e.g. an image of a ‘5’ that is similar to a ‘6’.
- **Near-OOD:** Near-OOD represents a label shift where label space is different between ID and OOD data. It has high epistemic uncertainty arising from the classifier’s limited information on unseen data. We use samples from unseen classes of the training distribution as near-OOD.
- **Far-OOD:** Far-OOD represents data distribution that is significantly different from the training distribution. It has high epistemic uncertainty arising from mismatch between different data distributions.

Much of the earlier work focuses on threshold-based detectors that use information from a pre-trained DNN to identify OOD samples [15, 19, 24, 68, 21]. Such methods focus on far-OOD detection and often do not address the overconfidence problem in DNN. In another line of research, variants of Bayesian models [52, 9] and ensemble learning [22, 33] were explored to provide reliable uncertainty estimates. Recently, there is a shift towards designing generalizable DNN that provide robust uncertainty estimates in a single forward pass [65, 4, 48]. Such methods usually propose changes to the DNN architecture [62], training procedure [71] or loss functions [50] to encourage separation between ID and OOD data. Popular methods include,

*Equal contribution

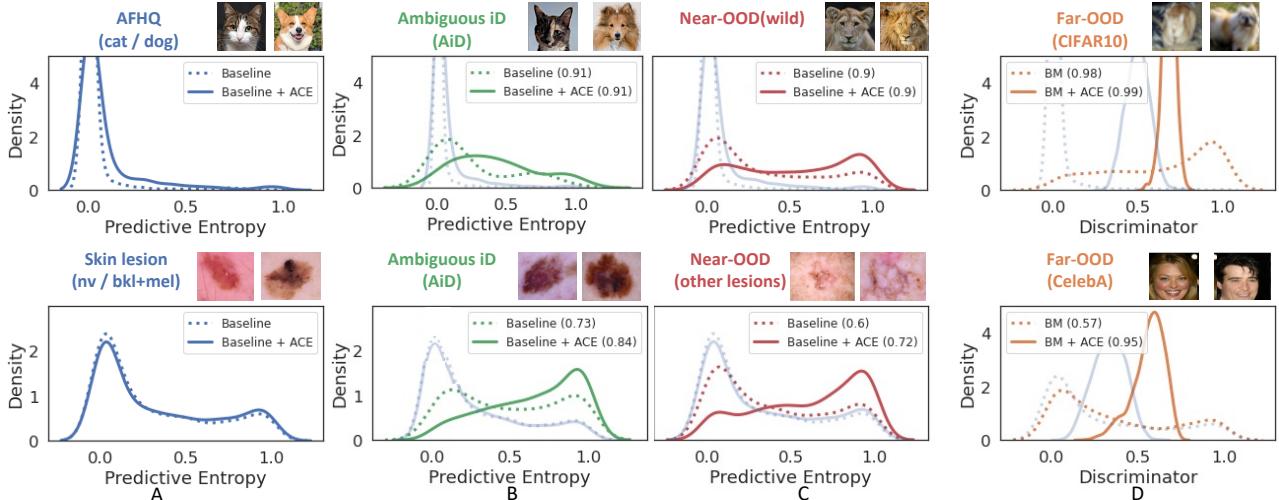


Figure 1. Comparison of the uncertainty estimates from the baseline, before (dotted line) and after (solid line) fine-tuning with augmentation by counterfactual explanation (ACE). The plots visualize the distribution of predicted entropy (columns A-C) from the classifier and density score from the discriminator (column D). The y-axis of this density plot is the probability density function whose value is meaningful only for relative comparisons between groups, summarized in the legend. **A**) visualizes the impact of fine-tuning on the **in-distribution (iD)** samples. A large overlap suggests minimum changes to classification outcome for iD samples. Next columns visualize change in the distribution for **ambiguous iD (AiD)** (**B**) and **near-OOD** samples (**C**). The peak of the distribution for AiD and near-OOD samples shifted right, thus assigning higher uncertainty and reducing overlap with iD samples. **D**) compares the density score from discriminator for **iD** (blue solid) and **far-OOD** (orange solid) samples. The overlap between the distributions is minimum, resulting in a high AUC-ROC for binary classification over uncertain samples and iD samples. Our method improved the uncertainty estimates across the spectrum.

training deterministic DNN with a distance-aware feature space [66, 42] and regularizing DNN training with a generative model that simulates OOD data [36]. However, these methods require a DNN model to be trained from scratch and are not compatible with an existing pre-trained DNN. Also, they may use auxiliary data to learn to distinguish OOD inputs [43].

Most of the DNN-based classification models are trained to improve accuracy on a test set. Accuracy only captures the proportion of samples that are on the correct side of the decision boundary. However, it ignores the relative distance of a sample from the decision boundary [31]. Ideally, samples closer to the boundary should have high uncertainty. The actual predicted value from the classifier should reflect this uncertainty via a low confidence score [25]. Conventionally, DNNs are trained on hard-label datasets to minimize a negative log-likelihood (NLL) loss. Such models tend to over-saturate on NLL and end-up learning very sharp decision boundaries [16, 49]. The resulting classifiers extrapolate over-confidently on ambiguous, near boundary samples, and the problem amplifies as we move to OOD regions [8].

In this paper, we propose to mitigate the overconfidence problem of a pre-trained DNN by fine-tuning it with augmentations derived from a counterfactual explainer (ACE). We derived counterfactuals using a progressive counterfactual explainer (PCE) that create a series of perturbations of an input image, such that the classification decision is changed to a different class [58, 34]. PCE is trained to

generate on-manifold samples in the regions between the classes. These samples along with soft labels that mimics their distance from the decision boundary, are used to fine-tuned the classifier. We hypothesis that fine-tuning on such data would broaden the classifier’s decision boundary. Our empirical results show the fine-tuned classifier exhibits better uncertainty quantification over ambiguous-iD and OOD samples. Our contributions are as follows: (1) We present a novel strategy to fine-tune an existing *pre-trained* DNN using ACE, to improve its uncertainty estimates. (2) We proposed a refined architecture to generate counterfactual explanations that takes into account continuous condition and multiple target classes. (3) We used the discriminator of our GAN-based counterfactual explainer as a selection function to reject far-OOD samples. (4) The fine-tuned classifier with rejection head, successfully captures uncertainty over ambiguous-iD and OOD samples, and also exhibits better robustness to popular adversarial attacks.

2. Method

In this paper, we consider a pre-trained DNN classifier, f_θ , with good prediction accuracy but sub-optimal uncertainty estimates. We assume f_θ is a differentiable function and we have access to its gradient with respect to the input, $\nabla_x f_\theta(\mathbf{x})$, and to its final prediction outcome $f_\theta(\mathbf{x})$. We also assume access to either the training data for f_θ , or an equivalent dataset with competitive prediction accuracy. We further assume that the training dataset for f_θ has hard labels $\{0, 1\}$ for all the classes.

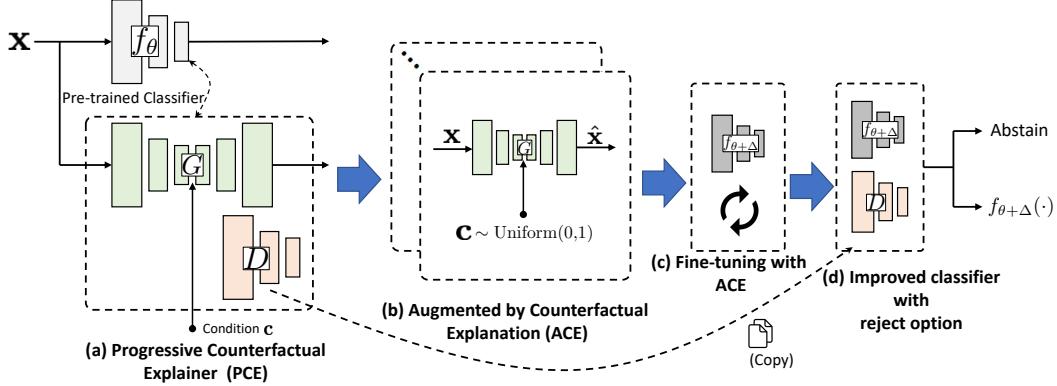


Figure 2. (a) Given a *pre-trained* classifier f_θ , we learn a c-GAN based progressive counterfactual explainer (PCE) $G(\mathbf{x}, \mathbf{c})$, while keeping f_θ fixed. (b) The trained PCE creates counterfactually augmented data. (c) A combination of original training data and augmented data is used to fine-tune the classifier, $f_{\theta+\Delta}$. (d) The discriminator from PCE serves as a selection function to detect and reject OOD data.

Our goal is to improve the pre-trained classifier f_θ such that the revised model provides better uncertainty estimates, while retaining its original predictive accuracy. To enable this, we follow a two step approach. First, we *fine-tune* f_θ on counterfactually augmented data. The fine-tuning helps in widening the classification boundary of f_θ , resulting in improved uncertainty estimates on ambiguous and near-OOD samples. Second, we use a density estimator to identify and reject far-OOD samples.

We adapted previously proposed PCE [58] to generate counterfactually augmented data. We improved the existing implementations of PCE, by adopting a StyleGANv2-based backbone for the conditional-GAN in PCE. This allows using continuous vector $f_\theta(\mathbf{x})$ as condition for conditional generation. Further, we used the discriminator of cGAN as a *selection function* to abstain revised $f_{\theta+\Delta}$ from making prediction on far-OOD samples (see Fig. 2).

Notation: The classification function is defined as $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^K$, where θ represents model parameters. The training dataset for f_θ is defined as $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, where $\mathbf{x} \in \mathcal{X}$ represents an input space and $y \in \mathcal{Y} = \{1, 2, \dots, K\}$ is a label set over K classes. The classifier produces point estimates to approximate the posterior probability $\mathbb{P}(y|\mathbf{x}, \mathcal{D})$.

2.1. Progressive Counterfactual Explainer (PCE)

We designed the PCE network to take a query image ($\mathbf{x} \in \mathbb{R}^d$) and a desired classification outcome ($\mathbf{c} \in \mathbb{R}^K$) as input, and create a perturbation of a query image ($\hat{\mathbf{x}}$) such that $f_\theta(\hat{\mathbf{x}}) \approx \mathbf{c}$. Our formulation, $\hat{\mathbf{x}} = G(\mathbf{x}, \mathbf{c})$ allows us to use \mathbf{c} to traverse through the decision boundary of f_θ from the original class to a counterfactual class. Following previous work [34, 58, 59], we design the PCE to satisfy the following three properties:

1. **Data consistency:** The perturbed image, $\hat{\mathbf{x}}$ should be realistic and should resemble samples in \mathcal{X} .

2. **Classifier consistency:** The perturbed image, $\hat{\mathbf{x}}$ should produce the desired output from the classifier f_θ i.e. $f_\theta(G(\mathbf{x}, \mathbf{c})) \approx \mathbf{c}$.
3. **Self consistency:** Using the original classification decision $f_\theta(\mathbf{x})$ as condition, the PCE should produce a perturbation that is very similar to the query image, i.e. $G(G(\mathbf{x}, \mathbf{c}), f_\theta(\mathbf{x})) = \mathbf{x}$ and $G(\mathbf{x}, f_\theta(\mathbf{x})) = \mathbf{x}$.

Data Consistency: We formulate the PCE as a cGAN that learns the underlying data distribution of the input space \mathcal{X} without an explicit likelihood assumption. The GAN model comprised of two networks – the generator $G(\cdot)$ and the discriminator $D(\cdot)$. The $G(\cdot)$ learns to generate fake data, while the $D(\cdot)$ is trained to distinguish between the real and fake samples. We jointly train G, D to optimize the following logistic adversarial loss [12],

$$\mathcal{L}_{\text{adv}}(D, G) = \mathbb{E}_{\mathbf{x}}[\log D(\mathbf{x}) + \log(1 - D(G(\mathbf{x}, \mathbf{c}))) \quad (1)$$

The earlier implementations of PCE [58], have a hard constraint of representing the condition \mathbf{c} as discrete variables. $f_\theta(\mathbf{x})$ is a continuous variable in range $[0, 1]$. We adapted StyleGANv2 [1] as the backbone of the cGAN. This formulation allow us to use $\mathbf{c} \in \mathbb{R}^K$ as condition.

We formulate the generator as $G(\mathbf{x}, \mathbf{c}) = g(e(\mathbf{x}), \mathbf{c})$, a composite of two functions, an image encoder $e(\cdot)$ and a conditional decoder $g(\cdot)$ [1]. The encoder function $e : \mathcal{X} \rightarrow \mathcal{W}^+$, learns a mapping from the input space \mathcal{X} to an extended latent space \mathcal{W}^+ . The detailed architecture is provided in Fig. 3. Further, we also extended the discriminator network $D(\cdot)$ to have auxiliary information from the classifier f_θ . Specifically, we concatenate the penultimate activations from the $f_\theta(\mathbf{x})$ with the penultimate activations from the $D(\mathbf{x})$, to obtain a revised representation before the final fully-connected layer of the discriminator. The detailed architecture is summarized in supplementary material (SM).

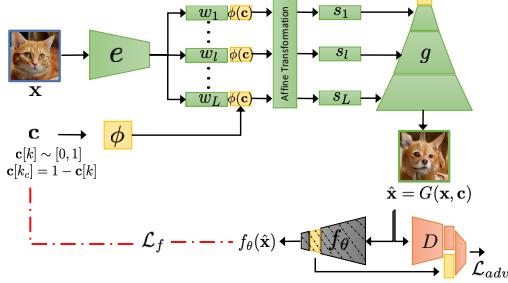


Figure 3. PCE: The encoder-decoder architecture to create counterfactual augmentation for a given query image. ACE: Given a query image, the trained PCE generates a series of perturbations that gradually traverse the decision boundary of f_θ from the original class to a counter-factual class, while still remaining plausible and realistic-looking.

We also borrow the concept of path-length regularization $\mathcal{L}_{\text{reg}}(G)$ from StyleGANv2 to enforce smoother latent space interpolations for the generator. $\mathcal{L}_{\text{reg}}(G) = \mathbb{E}_{\mathbf{w} \sim e(\mathbf{x}), \mathbf{x} \sim \mathcal{X}} (\|\mathbf{J}_\mathbf{w}^T \mathbf{x}\|_2 - a)^2$, where \mathbf{x} denotes random images from the training data, $\mathbf{J}_\mathbf{w}$ is the Jacobian matrix, and a is a constant that is set dynamically during optimization.

Classifier consistency: By default, GAN training is independent of the classifier f_θ . We add a classifier-consistency loss to regularize the generator and ensure that the actual classification outcome for the generated image $\hat{\mathbf{x}}$, is similar to the condition \mathbf{c} used for generation. We enforce classification-consistency by a KullbackLeibler (KL) divergence loss as follow[58, 59],

$$\mathcal{L}_f(G) = D_{KL}(f_\theta(\hat{\mathbf{x}}) \parallel \mathbf{c}) \quad (2)$$

Self consistency: We define the following reconstruction loss to regularize and constraint the Generator to preserve maximum information between the original image \mathbf{x} and its reconstruction $\bar{\mathbf{x}}$,

$$\mathcal{L}(\mathbf{x}, \bar{\mathbf{x}}) = \|\mathbf{x} - \bar{\mathbf{x}}\|_1 + \|e(\mathbf{x}) - e(\bar{\mathbf{x}})\|_1 \quad (3)$$

Here, first term is an L1 distance loss between the input and the reconstructed image, and the second term is a style reconstruction L1 loss adapted from StyleGANv2 [1]. We minimize this loss to satisfy the identify constraint on self reconstruction using $\bar{\mathbf{x}}_{\text{self}} = G(\mathbf{x}, f_\theta(\mathbf{x}))$. We further insure that the PCE learns a reversible perturbation by recovering the original image from a given perturbed image $\hat{\mathbf{x}}$ as $\bar{\mathbf{x}}_{\text{cyclic}} = G(\hat{\mathbf{x}}, f_\theta(\mathbf{x}))$, where $\hat{\mathbf{x}} = G(\mathbf{x}, \mathbf{c})$ with some condition \mathbf{c} . Our final reconstruction loss is defined as,

$$\mathcal{L}_{\text{rec}}(G) = \mathcal{L}(\mathbf{x}, \bar{\mathbf{x}}_{\text{self}}) + \mathcal{L}(\mathbf{x}, \bar{\mathbf{x}}_{\text{cyclic}}) \quad (4)$$

Objective function: Finally, we trained our model in an end-to-end fashion to learn parameters for the two networks,

while keeping the classifier f_θ fixed. Our overall objective function is

$$\begin{aligned} \min_G \max_D \lambda_{\text{adv}} (\mathcal{L}_{\text{adv}}(D, G) + \mathcal{L}_{\text{reg}}(G)) \\ + \lambda_f \mathcal{L}_f(G) + \lambda_{\text{rec}} \mathcal{L}_{\text{rec}}(G), \end{aligned} \quad (5)$$

where, λ 's are the hyper-parameters to balance each of the loss terms.

2.2. Augmentation by Counterfactual Explanation

Given a query image \mathbf{x} , the trained PCE generates a series of perturbations of \mathbf{x} that gradually traverse the decision boundary of f_θ from the original class to a counter-factual class, while still remaining plausible and realistic-looking. We modify \mathbf{c} to represent different steps in this traversal. We start from a high data-likelihood region for original class k ($\mathbf{c}[k] \in [0.8, 1.0]$), walk towards the decision hyper-plane ($\mathbf{c}[k] \in [0.5, 0.8]$), and eventually cross the decision boundary ($\mathbf{c}[k] \in [0.2, 0.5]$) to end the traversal in a high data-likelihood region for the counterfactual class k_c ($\mathbf{c}[k] \in [0.0, 0.2]$). Accordingly, we set $\mathbf{c}[k_c] = 1 - \mathbf{c}[k]$.

Ideally, the predicted confidence from NN should be indicative of the distance from the decision boundary. Samples that lies close to the decision boundary should have low confidence, and confidence should increase as we move away from the decision boundary. We used \mathbf{c} as a pseudo indicator of confidence to generate synthetic augmentation. Our augmentations are essentially showing how the query image \mathbf{x} should be modified to have low/high confidence.

To generate counterfactual augmentations, we randomly sample a subset of real training data as $\mathcal{X}_r \subset \mathcal{X}$. Next, for each $\mathbf{x} \in \mathcal{X}_r$ we generate multiple augmentations ($\hat{\mathbf{x}} = G(\mathbf{x}, \mathbf{c})$) by randomly sampling $\mathbf{c}[k] \in [0, 1]$. We used \mathbf{c} as soft label for the generate sample while fine-tuning the f_θ . The \mathcal{X}_c represents our pool of generated augmentation images. Finally, we create a new dataset by randomly sampling images from \mathcal{X} and \mathcal{X}_c . We fine-tune the f_θ on this new dataset, for only a few epochs, to obtain a revised classifier given as $f_{\theta+\Delta}$. In our experiments, we show that the revised decision function $f_{\hat{\theta}}$ provides improved confidence estimates for AiD and near OOD samples and demonstrate robustness to adversarial attacks, as compared to given classifier f_θ .

2.3. Discriminator as a Selection Function

A selection function $g : \mathcal{X} \rightarrow \{0, 1\}$ is an addition head on top of a classifier that decides when the classifier should abstain from making a prediction. We propose to use the discriminator network $D(\mathbf{x})$ as a selection function for f_θ . Upon the convergence of the PCE training, the generated samples resemble the in-distribution training data. Far-OOD samples are previously unseen samples which are

very different from the training input space. Hence, $D(\cdot)$ can help in detecting such samples. Our final improved classification function is represented as follow,

$$(f, D)(\mathbf{x}) = \begin{cases} f_{\theta+\Delta}(\mathbf{x}), & \text{if } D(\mathbf{x}) \geq h \\ \text{Abstain}, & \text{otherwise} \end{cases} \quad (6)$$

where, $f_{\theta+\Delta}$ is the fine-tuned classifier and $D(\cdot)$ is a discriminator network from the PCE which serves as a selection function that permits f to make prediction if $D(\mathbf{x})$ exceeds a threshold h and abstain otherwise.

3. Related Work

Uncertainty estimation in pre-trained DNN models: Much of the prior work focused on deriving uncertainty measurements from a pre-trained DNN output [19, 15, 39, 43], feature representations [41, 37] or gradients [24]. Such methods use a threshold-based scoring function to identify OOD samples. The scoring function is derived from softmax confidence scores [19], scaled logit [15, 41], energy-based scores [43, 68] or gradient-based scores [24]. These methods help in identifying OOD samples but did not address the over-confidence problem of DNN, that made identifying OOD non-trivial in the first place [18, 54]. We propose to mitigate the over-confidence issue by fine-tuning the pre-trained classifier using ACE. Further, we used a hard threshold on the density score provided by the discriminator of the GAN-generator, to identify OOD samples.

DNN designs for improved uncertainty estimation: The Bayesian neural networks are the gold standard for reliable uncertainty quantification [52]. Multiple approximate Bayesian approaches have been proposed to achieve tractable inference and to reduce computational complexity [14, 2, 28, 9]. Popular non-Bayesian methods include deep ensembles [33] and their variant [22, 10]. However, most of these methods are computationally expensive and requires multiple passes during inference. An alternative approach is to modify DNN training [63, 71, 67], loss function [50], architecture [62, 42, 11] or end-layers [66, 21] to support improved uncertainty estimates in a single forward-pass. Further, methods such as DUQ [66] and DDU [48] proposed modifications to enable the separation between aleatoric and epistemic uncertainty. Unlike these methods, our approach improves the uncertainty estimates of any existing pre-trained classifier, without changing its architecture or training procedure. We used the discriminative head of the fine-tuned classifier to capture aleatoric uncertainty and the density estimation from the GAN-generator to capture epistemic uncertainty.

Uncertainty estimation using GAN: A popular technique to fix an over-confident classifier is to regularize the model with an auxiliary OOD data which is either realistic [20, 46, 55, 4, 40] or is generated using GAN [56, 36,

45, 70, 57]. Such regularization helps the classifier to assign lower confidence to anomalous samples, which usually lies in the low-density regions. Defining the scope of OOD a-priori is generally hard and can potentially cause a selection bias in the learning. Alternative approaches resort to estimating in-distribution density [61]. Our work fixed the scope of GAN-generation to counterfactual generation. Rather than merging the classifier and the GAN training, we train the GAN in a post-hoc manner to explain the decision of an existing classifier. This strategy defines OOD in the context of pre-trained classifier’s decision boundary. Previously, training with CAD have shown to improved generalization performance on OOD samples [27]. However, this work is limited to Natural Language Processing, and requires human intervention while curating CAD [26]. In contrast, we train a GAN-based counterfactual explainer [59, 34] to derive CAD.

4. Experiment

We consider four classification problems, in increasing level of difficulty:

1. AFHQ [5]: We consider binary classification over well separated classes, cat vs dog.
2. Dirty MNIST [48]: We consider multi-class classification over hand-written digits 0-6. The dataset is a combination of original MNIST [35] and simulated samples from a variational decoder. The samples are generated by combining latent representation of different digits, to simulate ambiguous samples, with multiple plausible labels [48].
3. CelebA [44]: We consider a multi-label classification setting over ‘young’ and ‘smiling’ attributes. Without age labels, identifying ‘young’ faces is a challenging task.
4. Skin lesion (HAM10K) [64]: We consider a binary classification to separate Melanocytic nevus (nv) from Melanoma (mel) and Benign Keratosis (bkl) lesions. Skin lesion classification is a challenging task as different lesions may exhibit similar features [51].

Architecture details: We consider state-of-the-art DenseNet [23] architecture for the baseline. The *pre-trained* DenseNet model followed the training procedures as described in [23]. In order to keep the architecture and training procedure of PCE simple, we consider the default training parameters from [1] for training the StyleGANv2. This encourages reproducibility as we didn’t do hyper-parameter tuning for each dataset and classification model. For training StyleGANv2, we use a randomly sampled subset ($\sim 50\%$) of the baseline model’s training data. For multi-class classification, we consider all pairs

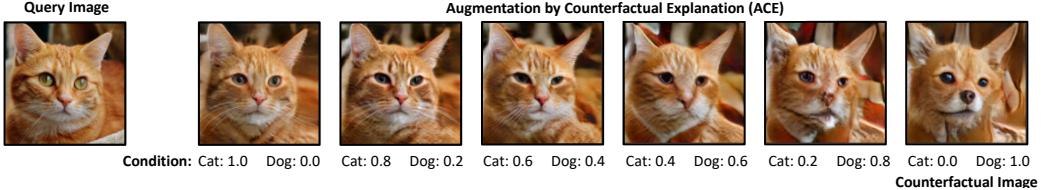


Figure 4. An example of counterfactually generated data by the progressive counterfactual explainer (PCE). More examples are provided in the Supplementary Material.

of classes while creating counterfactual augmentations. For fine-tuning the baseline, we create a new dataset with 30% counterfactually generated samples and 70% real samples from the original training set. All the results are reported on the test set of the baseline. In all our experiments, we used $\lambda_{adv} = 10$, $\lambda_{rec} = 100$, $\lambda_f = 10$, and $h = 0.5$.

Comparison methods: Our baseline is a standard DNN classifier f_θ trained with cross-entropy loss. For baseline and its post-hoc variant with temperature-scaling (**TS**), we used threshold over predictive entropy (PE) to identify OOD. PE is defined as $-\sum [f_\theta(\mathbf{x})]_k \log [f_\theta(\mathbf{x})]_k$. Next, we compared against following five methods: **mixup**: baseline model with mixup training using $\alpha = 0.2$ [71]; deterministic uncertainty quantification (**DUQ**) [66]: baseline model with radial basis function as end-layer; **DDU**: that use the closest kernel distance to quantify uncertainties; **MC Dropout** (with 20 dropout samples) [9]; and five independent runs of baseline as **5-Ensemble** [33]. The ensemble approaches are an upper bound for UQ.

4.1. Identifying AiD samples

We do not have access to ground truth labels marking the samples that are AiD. Hence, we used the PE estimates from an MC Dropout classifier to obtain pseudo-ground truth for AiD classification. Specifically, we sort the test set using PE and consider the top 5 to 10% samples as AiD. In Fig. 1, we qualitatively compare the PE distribution from the given baseline and its fine-tuned version (baseline + ACE). Fine-tuning resulted in minor changes to the PE distribution of the iD samples (Fig. 1.A). We observe a significant separation in the PE distribution of AiD samples and the rest of the test set (Fig. 1.B), even on the baseline. This suggests that the PE correctly captures the aleatoric uncertainty.

Table 1 compares our model to several baselines. We report the test set accuracy, the AUC-ROC for the binary task of identifying AiD samples and the true negative rate (TNR) at 95% true positive rate (TPR) (TNR@TPR95), which simulates an application requirement that the recall of in-distribution data should be 95% [21]. For all metrics higher value is better. Our model outperformed other deterministic models in identifying AiD samples with a high AUC-ROC and TNR@TPR95 across all datasets.

4.2. Detecting OOD samples

We consider two tasks to evaluate the model’s OOD detection performance. First, a standard OOD task where OOD samples are derived from a separate dataset. Second, a difficult near-OOD detection task where OOD samples belongs to novel classes from the same dataset, which are not seen during training. We consider the following OOD datasets:

1. AFHQ [5]: We consider “wild” class from AFHQ to define near-OOD samples. For the far-OOD detection task, we use the CelebA dataset, and also cat/dog images from CIFAR10 [32].
2. Dirty MNIST [48]: We consider digits 7-9 as near-OOD samples. For far-OOD detection, we use SVHN [53] and fashion MNIST [69] datasets.
3. CelebA [44]: We consider images of kids in age-group: 0-11 from the UTKFace [72] dataset to define the near-OOD samples. For far-OOD detection task, we use the AFHQ and CIFAR10 datasets.
4. Skin lesion (HAM10K) [64]: We consider samples from lesion types: Actinic Keratoses and Intraepithelial Carcinoma (akiec), Basal Cell Carcinoma (bcc), Dermatofibroma (df) and Vascular skin lesions (vasc) as near-OOD. For far-OOD, we consider CelebA and an additional simulated dataset with different skin textures/tones.

In Fig. 1, we observe much overlap between the PE distribution of the near-OOD samples and in-distribution samples in Fig. 1.C. Further, in Fig. 1.D, we see that our model successfully disentangles OOD samples from the in-distribution samples by using density estimates from the discriminator of the PCE. In Table 2, we report the AUC-ROC and TNR@TPR95 scores on detecting the two types of OOD samples. We first use the discriminator from the PCE to detect far-OOD samples. The discriminator achieved near-perfect AUC-ROC for detecting far-OOD samples. We used the PE estimates from the fine-tuned model (baseline + ACE) to detect near-OOD samples. Overall our model outperformed other methods on both near and far-OOD detection tasks with high TNR@TPR95.

Table 1. Performance of different methods on identifying **ambiguous in-distribution (AiD)** samples. For all metrics, higher is better. The best results from the methods that require a single forward pass at inference time are highlighted.

Train Dataset	Method/ Model	Test-Set Accuracy	Identifying AiD	
			AUC-ROC	TNR@TPR95
AFHQ	Baseline	99.44±0.02	0.87±0.04	48.93±10
	Baseline+TS [17]	99.45±0.00	0.85±0.07	48.77±9.8
	Baseline+TS+ODIN [39]	99.45±0.00	0.85±0.06	35.72±1.26
	Baseline+energy [43]	99.44±0.02	0.87±0.06	49.00±1.64
	Mixup [71]	99.02±0.10	0.80±0.05	35.66±6.7
	DUQ [66]	94.00±1.05	0.67±0.01	26.15±4.5
	DDU [48]	97.66±1.10	0.74±0.02	19.65±4.5
	Baseline+ACE	99.52±0.21	0.91±0.02	50.75±3.9
Dirty MNIST	MC Dropout [9]	98.83±1.12	0.87±0.04	51.56±1.2
	5-Ensemble [33]	99.79±0.01	0.98±0.01	51.93±2.7
	Baseline	95.68±0.02	0.96±0.00	28.5±2.3
	Baseline+TS [17]	95.74±0.02	0.94±0.01	27.90±1.3
	Baseline+TS+ODIN [39]	95.74±0.02	0.79±0.03	13.25±4.88
	Baseline+energy [43]	95.68±0.02	0.80±0.03	17.60±0.55
	Mixup [71]	94.66±0.16	0.94±0.02	25.78±2.1
	DUQ [66]	89.34±0.44	0.67±0.01	23.89±1.2
CelebA	DDU [48]	93.52±1.12	0.65±0.12	20.78±4.0
	Baseline+ACE	95.36±0.45	0.86±0.01	34.12±2.6
	MC Dropout [9]	89.50±1.90	0.75±0.07	36.10±1.8
	5-Ensemble [33]	95.90±0.12	0.98±0.02	34.87±3.4
	Baseline	89.36±0.96	0.73±0.01	17.18±1.6
	Baseline+TS [17]	89.33±0.01	0.72±0.02	17.21±1.5
	Baseline+TS+ODIN [39]	89.33±0.01	0.57±0.01	6.34±0.38
	Baseline+energy [43]	89.36±0.96	0.57±0.28	4.87±0.32
Skin-Lesion (HAM10K)	Mixup [71]	89.04±0.47	0.74±0.02	15.09±1.9
	DUQ [66]	71.75±0.01	0.65±0.01	14.20±1.0
	DDU [48]	70.15±0.02	0.67±0.06	11.39±0.4
	Baseline+ACE	86.8±0.79	0.74±0.06	22.36± 2.3
	MC Dropout [9]	89.86±0.33	0.73±0.03	19.78±0.7
	5-Ensemble [33]	90.76±0.00	0.84±0.11	17.79±0.6
	Baseline	85.88±0.75	0.82±0.06	20.52±3.7
	Baseline+TS [17]	86.27±0.40	0.84±0.03	23.34±2.8
HAM10K	Baseline+TS+ODIN [39]	86.27±0.40	0.78±0.01	15.87±4.33
	Baseline+energy [43]	85.88±0.75	0.77±0.12	18.40±0.51
	Mixup [71]	85.81±0.61	0.84±0.04	31.29±7.0
	DUQ [66]	75.47±5.36	0.81±0.02	30.12±4.4
	DDU [48]	75.84±2.34	0.79±0.03	26.12±6.6
	Baseline+ACE	81.21±1.12	0.84±0.05	71.60±3.8
	MC Dropout [9]	84.90±1.17	0.85±0.06	43.78±1.9
	5-Ensemble [33]	87.89±0.13	0.86±0.02	40.49±5.1

4.3. Robustness to Adversarial Attacks

We compared the baseline model before and after fine-tuning (baseline + ACE) in their robustness to three adversarial attacks: Fast Gradient Sign Method (FGSM) [13], Carlini-Wagner (CW) [3], and DeepFool [47].

For each attack setting, we transformed the test set into an adversarial set. In Fig. 5, we report the AUC-ROC over the adversarial set as we gradually increase the magnitude of the attack. For FGSM, we use the maximum perturbation (ϵ) to specify the attack’s magnitude. For CW, we gradually increase the number of iterations to achieve a higher magnitude attack. We set box-constraint parameter as $c = 1$, learning rate $\alpha = 0.01$ and confidence $\kappa = 0, 5$. For DeepFool ($\eta = 0.02$), we show results on the best attack. Our improved model (baseline + ACE) consistently out-performed the baseline model in test AUC-ROC, thus showing an improved robustness to all three attacks.

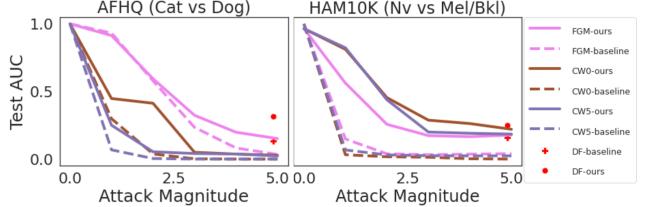


Figure 5. Plots comparing baseline model before and after fine-tuning (ACE) for different magnitudes of adversarial attack. The figure shows three different attacks – FGSM [13], CW [3], DeepFool [47], on three different datasets – HAM10K, AFHQ, MNIST. The x-axis denotes maximum perturbation (ϵ) for FGSM, and iterations in multiples of 10 for CW and DeepFool. Attack magnitude of 0 indicates no attack. For CW we used $\kappa = 0$ and 5. (All results are reported on the test-set of the classifier).

5. Conclusion

We propose a novel application of counterfactual explanations in improving the uncertainty quantification of a

Table 2. OOD detection performance for different baselines. **Near-OOD** represents label shift, with samples from the unseen classes of the same dataset. **Far-OOD** samples are from a separate dataset. The numbers are averaged over five runs.

Train Dataset	Method	Near-OOD (Wild)		Far-OOD (CIFAR10)		Far-OOD (CelebA)	
		AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95
AFHQ	Baseline	0.88±0.04	47.40±5.2	0.95±0.04	73.59±9.4	0.95±0.03	70.69±8.9
	Baseline+TS [17]	0.88±0.03	45.53±9.8	0.95±0.04	71.77±8.9	0.95±0.03	65.89±8.3
	Baseline+TS+ODIN [39]	0.87±0.05	45.02±1.51	0.95±0.05	69.42±2.38	0.95±0.03	67.18±2.16
	Baseline+energy [43]	0.88±0.03	47.77±1.10	0.94±0.05	72.68±2.69	0.96±0.04	74.75±2.89
	Mixup [71]	0.86±0.06	53.83±6.8	0.82±0.11	57.01±8.6	0.88±0.13	70.51±9.8
	DUQ [66]	0.78±0.05	20.98±2.0	0.67±0.59	16.23±1.5	0.66±0.55	15.34±2.6
	DDU [48]	0.83±0.02	23.19±2.6	0.90±0.02	32.98±10	0.75±0.02	10.32±5.6
	Baseline+ACE	0.89±0.03	51.39±4.4	0.98±0.02	88.71±5.7	0.97±0.03	88.87±9.8
	MC-Dropout [9]	0.84±0.09	30.78±2.9	0.94±0.02	73.59±2.1	0.95±0.02	71.23±1.9
	5-Ensemble [33]	0.99±0.01	65.73±1.2	0.97±0.02	89.91±0.9	0.99±0.01	92.12±0.7
		Near-OOD (Digits 7-9)		Far-OOD (SVHN)		Far-OOD (fMNIST)	
		AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95
Dirty MNIST	Baseline	0.86±0.04	28.23±2.9	0.75±0.15	51.98±0.9	0.87±0.02	58.12±1.5
	Baseline+TS [17]	0.86±0.01	30.12±2.1	0.73±0.07	48.12±1.5	0.89±0.01	61.71±2.8
	Baseline+TS+ODIN [39]	0.83±0.04	34.13±12.07	0.77±0.13	21.59±19.62	0.89±0.02	46.43±4.31
	Baseline+energy [43]	0.87±0.04	40.30±1.05	0.86±0.12	43.92±2.30	0.91±0.02	62.10±5.17
	Mixup [71]	0.86±0.02	35.46±1.0	0.95±0.03	65.12±3.1	0.94±0.05	66.00±0.8
	DUQ [66]	0.78±0.01	15.26±3.9	0.73±0.03	45.23±1.9	0.75±0.03	50.29±3.1
	DDU [48]	0.67±0.07	10.23±0.9	0.68±0.04	39.31±2.2	0.85±0.02	53.76±3.7
	Baseline+ACE	0.94±0.02	37.23±1.9	0.98±0.02	67.88±3.1	0.97±0.02	70.71±1.1
	MC-Dropout [9]	0.97±0.02	40.89±1.5	0.95±0.01	62.12±5.7	0.93±0.02	65.01±0.7
	5-Ensemble [33]	0.98±0.02	42.17±1.0	0.82±0.03	55.12±2.1	0.94±0.01	64.19±4.2
		Near-OOD (Kids)		Far-OOD (AFHQ)		Far-OOD (CIFAR10)	
		AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95
CelebA	Baseline	0.84±0.02	1.25±0.1	0.86±0.03	88.57±0.9	0.79±0.02	29.01±5.1
	Baseline+TS [17]	0.82±0.04	1.24±0.1	0.87±0.06	88.75±0.9	0.78±0.04	29.01±5.1
	Baseline+TS+ODIN [39]	0.65±0.01	8.75±2.21	0.55±0.01	23.03±0.16	0.54±0.01	5.00±0.07
	Baseline+energy [43]	0.76±0.51	9.40±0.01	0.94±0.08	32.08±1.70	0.85±0.76	17.10±0.72
	Mixup [71]	0.82±0.08	22.18±2.7	0.95±0.02	82.96±2.5	0.79±0.13	30.54±1.3
	DUQ [66]	0.80±0.03	14.68±3.1	0.72±0.07	26.62±7.7	0.86±0.04	28.70±4.1
	DDU [48]	0.73±0.15	7.9±0.4	0.74±0.13	8.18±0.4	0.81±0.15	25.45±1.4
	Baseline+ACE	0.87±0.03	34.37±2.5	0.96±0.01	96.35±2.5	0.92±0.05	63.51±1.5
	MC-Dropout [9]	0.70±0.10	25.62±1.7	0.86±0.1	91.72±7.5	0.74±0.12	64.79±1.8
	5-Ensemble [33]	0.93±0.03	10.35±0.2	0.99±0.0	98.31±1.2	0.92±0.10	61.88±1.2
		Near-OOD (other lesions)		Far-OOD (CelebA)		Far-OOD (Skin-texture)	
		AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95
Skin Lesion	Baseline	0.67±0.04	8.70±2.5	0.66±0.06	10.00±3.6	0.65±0.10	5.91±2.8
	Baseline+TS [17]	0.67±0.05	8.69±2.0	0.63±0.06	9.24±4.3	0.68±0.07	5.70±3.2
	Baseline+TS+ODIN [39]	0.68±0.01	9.43±0.33	0.67±0.07	11.32±4.66	0.68±0.07	6.60±0.29
	Baseline+energy [43]	0.70±0.04	10.85±0.08	0.70±0.14	7.90±0.29	0.65±0.20	2.83±1.33
	Mixup [71]	0.67±0.01	8.52±2.8	0.64±0.08	10.21±4.0	0.72±0.05	5.26±3.1
	DUQ [66]	0.67±0.04	3.12±1.8	0.89±0.09	11.89±2.5	0.64±0.03	4.8±1.5
	DDU [48]	0.65±0.03	3.45±1.9	0.75±0.04	15.45±2.9	0.71±0.05	4.19±1.3
	Baseline+ACE	0.72±0.04	10.99±2.8	0.97±0.02	66.77±1.4	0.96±0.03	95.83±5.0
	MC-Dropout [9]	0.67±0.05	9.45±3.9	0.80±0.07	30.00±3.2	0.56±0.03	10.87±2.3
	5-Ensemble [33]	0.88±0.01	11.23±1.7	0.91±0.03	27.89±5.9	0.76±0.02	17.89±3.5

pre-trained DNN. We improved upon the existing work on counterfactual explanations, by proposing a StyleGANv2-based backbone. Fine-tuning on augmented data, with soft labels helps in improving the decision boundary and

the fine-tuned model, combined with the discriminator of the PCE can successfully capture uncertainty over ambiguous samples, unseen near-OOD samples with label shift and far-OOD samples from independent datasets. We out-

performed state-of-the-art methods for uncertainty quantification on four datasets, and our improved model also exhibits robustness to adversarial attacks.

References

- [1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.
- [2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *32nd International Conference on International Conference on Machine Learning*, page 1613–1622, 2015.
- [3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.
- [4] Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. Atom: Robustifying out-of-distribution detection using outlier mining. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2021.
- [5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.
- [6] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 2017 542:7639, 542(7639):115–118, 1 2017.
- [7] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. *CoRR*, abs/1804.05132, 2018.
- [8] Yarin Gal. Uncertainty in deep learning. 2016.
- [9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [10] Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. *32nd International Conference on Neural Information Processing Systems*, page 8803–8812, 2018.
- [11] Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. *ICML*, 2019.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [14] Alex Graves. Practical variational inference for neural networks. *Advances in Neural Information Processing Systems*, 2011.
- [15] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- [16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- [17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- [18] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–50, 2019.
- [19] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.
- [20] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *International Conference on Learning Representations*, 2019.
- [21] Y. C. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [22] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. *5th International Conference on Learning Representations, ICLR*, 2017.
- [23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [24] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 2021.
- [25] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.*, 110:457–506, 2021.
- [26] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *International Conference on Learning Representations*, 2020.
- [27] Divyansh Kaushik, Amrit Rajagopal Setlur, Eduard H. Hovy, and Zachary Chase Lipton. Explaining the efficacy of counterfactually-augmented data. *International Conference on Learning Representations*, 2021.
- [28] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Advances in Neural Information Processing Systems*, 2015.

- [29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.
- [30] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. Risk Acceptance and Risk Communication.
- [31] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33, 2020.
- [32] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.
- [33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [34] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mossner. Explaining in style: Training a gan to explain a classifier in stylespace. *arXiv preprint arXiv:2104.13369*, 2021.
- [35] Yann LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [36] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *International Conference on Learning Representations*, 2018.
- [37] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 2018.
- [38] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7:17816, 2017.
- [39] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations*, 2018.
- [40] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *6th International Conference on Learning Representations ICLR*, 2018.
- [41] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [42] Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *ArXiv*, abs/2006.10108, 2020.
- [43] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.
- [44] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.
- [45] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [46] Sina Mohseni, Mandar Pitale, Jbs Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. *AAAI*, 2020.
- [47] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.
- [48] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty. *arXiv preprint arXiv:2102.11582*, 2021.
- [49] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33, 2020.
- [50] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet Kumar Dokania. Calibrating deep neural networks using focal loss. *ArXiv*, abs/2002.09437, 2020.
- [51] Franz Nachbar, Wilhelm Stoltz, Tanja Merkle, Armand B Cognetta, Thomas Vogt, Michael Landthaler, Peter Bilek, Otto Braun-Falco, and Gerd Plewig. The abcd rule of dermatoscopy: high prospective value in the diagnosis of doubtful melanocytic skin lesions. *Journal of the American Academy of Dermatology*, 30(4):551–559, 1994.
- [52] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- [53] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.
- [54] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.
- [55] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, 2021.
- [56] J. Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *NeurIPS*, 2019.
- [57] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and

- out-of-distribution detection with likelihood-based generative models. *International Conference on Learning Representations*, 2020.
- [58] Sumedha Singla, Brian Pollack, Junxiang Chen, and Kayhan Batmanghelich. Explanation by progressive exaggeration. *arXiv preprint arXiv:1911.00483*, 2019.
- [59] Sumedha Singla, Brian Pollack, Stephen Wallace, and Kayhan Batmanghelich. Explaining the black-box smoothly-a counterfactual approach. *arXiv preprint arXiv:2101.04230*, 2021.
- [60] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. *Uncertainty in Artificial Intelligence (UAI)*, 2018.
- [61] Akshayvarun Subramanya, Suraj Srinivas, and R. Venkatesh Babu. Confidence estimation in deep neural networks via density modelling. *ArXiv*, abs/1707.07013, 2017.
- [62] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 2021.
- [63] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [64] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.
- [65] Joost van Amersfoort, Lewis Smith, Andrew Jesson, Oscar Key, and Yarin Gal. On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*, 2021.
- [66] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. 2020.
- [67] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.
- [68] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don’t know? *Advances in Neural Information Processing Systems*, 2021.
- [69] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashionmnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.
- [70] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *ArXiv*, abs/2003.02977, 2020.
- [71] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- [72] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

6. Supplemental Material

6.1. Implementation Details

6.1.1 Dataset

We focus on improving classification models based on deep convolution neural networks (CNN) as most state-of-the-art performance models fall in this regime. In our experiments, we consider classification models trained on following datasets:

1. AFHQ [5]: Animal face high quality (AFHQ) dataset is a high resolution dataset of animal faces with 16K images from cat, dog and wild labels. In our experiments, we consider a multi-class classifier over cat and dog labels. We consider images with “wild” label as near-OOD. The classifier is trained at an image resolution of 256×256 .
2. Dirty MNIST [48]: The dataset is a combination of original MNIST [35] and simulated Ambiguous-MNIST dataset. Each sample in Ambiguous-MNIST is constructed by decoding a linear combination of latent representations of two different MNIST digits from a pre-trained VAE [29]. The training dataset of the classifier comprises of 60K clean-MNIST and 60K Ambiguous-MNIST samples, with one-hot labels. In our experiments, we consider classifier trained on seven classes over digits ‘0’ - ‘6’. We consider images from digits ‘7’ - ‘9’ as near-OOD samples. The original dataset consists of grayscale images of size 28×28 pixels. We consider a classification model trained on 64×64 resolution.
3. Skin lesion (HAM10K) [64]: The HAM10000 is a dataset of 100K dermatoscopic images of pigmented skin lesions. It contains seven different lesion types – Melanocytic Nevi (nv), Melanoma (mel), Benign Keratosis (bkl), Actinic Keratoses and Intraepithelial Carcinoma (akiec), Basal Cell Carcinoma (bcc), Dermatofibroma (df), Vascular skin lesions (vasc). In our experiments, we consider classifier trained to distinguish the majority class nv from mel and bkl. We consider images from rest of the lesions as near-OOD. The classifier is trained at an image resolution of 256×256 .
4. CelebA [44] : Celeb Faces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 binary attributes annotations per image. In our experiments, we consider a two-class classifier over attributes “Young” and “Smiling” trained on CelebA dataset. Our **AiD** samples comprises of middle-aged people who are arguably neither young nor old. To obtain such data, we use aleatoric uncertainty estimates

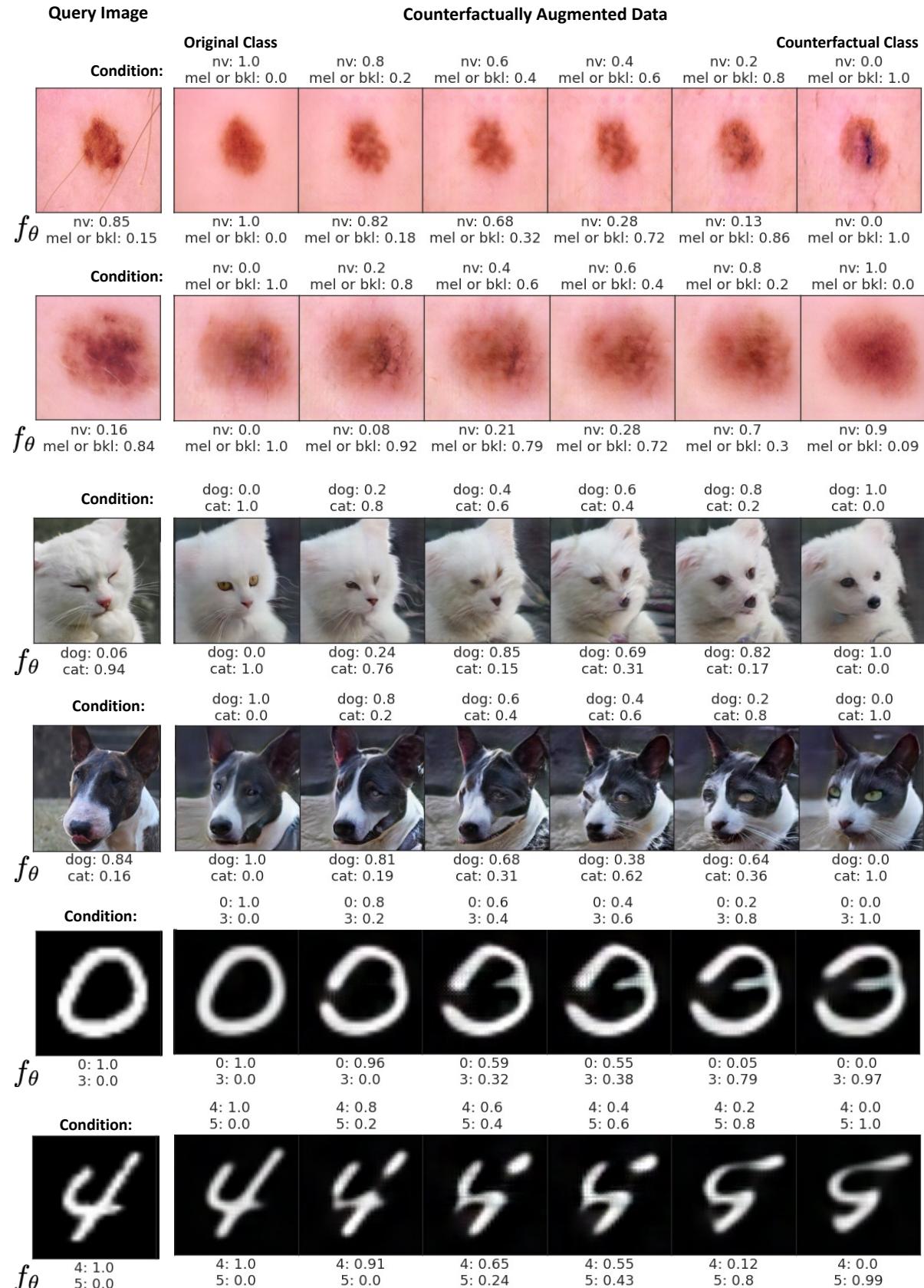


Figure 6. Examples of data augmentation using counterfactual explanations for different datasets.

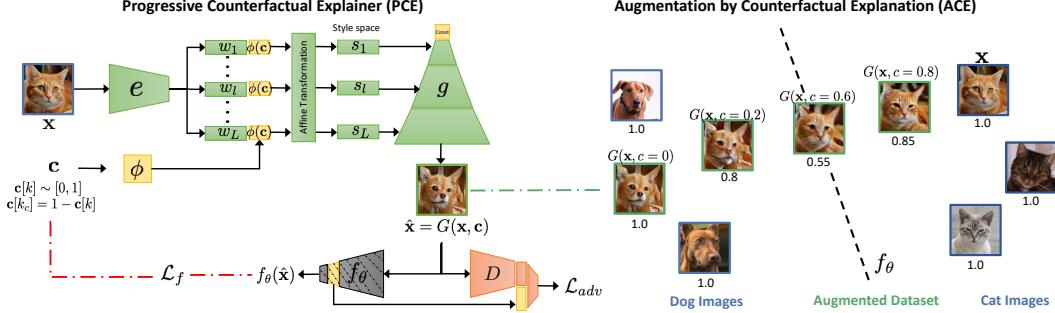


Figure 7. PCE: The encoder-decoder architecture to create counterfactual augmentation for a given query image. ACE: Given a query image, the trained PCE generates a series of perturbations that gradually traverse the decision boundary of f_θ from the original class to a counter-factual class, while still remaining plausible and realistic-looking.

from MC-Dropout averaged across 50 runs on test-set of CelebA. The classifier is trained at an image resolution of 256×256 . We center-crop the images as a pre-processing step.

6.1.2 Classification Model

We used DenseNet architecture as the classification model. In DenseNet, each layer implements a non-linear transformation based on composite functions such as Batch Normalization (BN), rectified linear unit (ReLU), pooling, or convolution. The resulting feature map at each layer is used as input for all the subsequent layers, leading to a highly convolved multi-level multi-layer non-linear convolutional neural network. We aim to improve such a model in a post-hoc manner without accessing the parameters learned by any layer or knowing the architectural details. Our proposed approach can be used for any DNN architecture.

6.2. Progressive Counterfactual Explainer

We formulate the progressive counterfactual explainer (PCE) as a composite of two functions, an image encoder $e(\cdot)$ and a conditional decoder ($g(\cdot)$) [1]. Our architecture for the conditional decoder is adapted from StyleGANv2 [1]. The image encoder converts the input image x into l different latent codes ($w_l \in \mathbb{R}^{512}$), for each of the L layers of the decoder. The decoder further transforms the layer-specific latent representation into a layer-specific style-vector as $s_l = A_l([w_l, \phi(c)])$ where, A_l is an affine transformation and $\phi(c)$ is an embedding for c . For training the StyleGANv2 decoder, we consider the default training parameters from [1]. For training the PCE, we use a randomly sampled subset ($\sim 50\%$) of the baseline training data. Given an input image, the predicted class k and a counterfactual class k_c , we initialize the condition c with all zeros and then set $c[k] \sim \text{Uniform}(0, 1)$ and $c[k_c] = 1 - c[k]$. In all our experiments, we used $\lambda_{adv} = 10$, $\lambda_{rec} = 100$ and $\lambda_f = 10$. Fig. 7 summarizes our architec-

ture.

For generating counterfactually augmented data, we first consider a randomly selected subset of real training data as $\mathcal{X}_r \in \mathcal{X}$. For each image in \mathcal{X}_r , we generate four augmented images by randomly selecting the $c[k]$. For each augmented image, we used the condition used to generate the image as the soft label while fine-tuning. Fig. 7 shows an example of our data augmentation. We denote the pool of the augmented images as \mathcal{X}_c . In Fig. 6, we show examples of counterfactual augmentation from different datasets.

For fine-tuning the given baseline with consider a combination of the original training dataset \mathcal{X} and the augmented data \mathcal{X}_c . We randomly selected a subset of samples from the two distributions and fine-tune the baseline for 5 to 10 epochs. We used the expected calibration error and the test-set accuracy to choose the final checkpoint. Our model does not require access to OOD or AiD dataset during fine-tuning. During evaluation we compute predicted entropy (PE) for original test-set and OOD samples and measure for a range of thresholds how well the two are separated. We report the AUC-ROC and the true negative rate (TNR) at 95% true positive rate (TPR) (TNR@TPR95) in our results (see Table 1 and 2).

6.3. Toy-Setup: Two-Moons

In this section, we demonstrate our method on a toy setup: the Two Moons dataset. We used the experimental set-up from DDU [48] for this experiment. We use scikit-learn’s datasets package to generate 2000 samples with a noise rate of 0.1. For baseline classification model, we use a 2-layer dense-layer architecture, with ReLU activation and batch normalization. The 2-D input data is projected to a 64-D latent space and then to 1D space to make final binary prediction. In Fig. 8.a, we show the uncertainty estimates (predicted entropy PE) from the baseline classifier. The baseline classifier is uncertain only along the decision boundary, and certain elsewhere (low PE).

Given the baseline classifier, we train a PCE to gener-

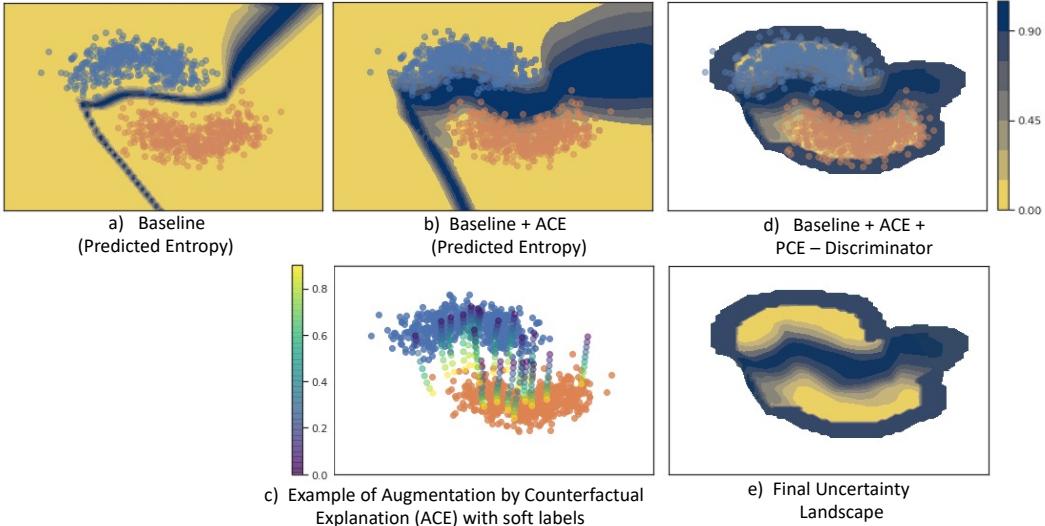


Figure 8. Uncertainty results on Two Moons dataset. Yellow indicates low uncertainty, while blue indicates uncertainty. a) The baseline classifier is uncertain only along the decision boundary, and certain elsewhere. b) Fine-tuning baseline model on ACE data improves uncertainty estimates near the decision boundary. c) An example of augmented data and corresponding soft labels. d) The discriminator from PCE rejects OOD samples, hence the rejected space have no uncertainty values (white color). e) The final uncertainty landscape, the improved classifier is certain on in-distribution regions and rejects OOD data.

ate augmented data. We use an encoder with two fully-connected layers that map 2-D input data to a 64-D latent space. The condition is also projected to a 64-D space and is concatenated to the output of the encoder. The decoder also have two fully-connected layers that maps the concatenated 128-D latent vector back to a 2-D input space. In Fig. 8.c, we show example of augmentation by counterfactual explanation (ACE). Given a query point, we generate series of augmented data by gradually changing the condition such that the decision of the baseline is flipped. The color of the dot represents the conditioned used to create the augmented sample. Next, we fine-tune the classifier using the original and the counterfactually augmented data. In Fig. 8.b, we show the PE estimates from the fine-tuned classifier (baseline + ACE). Fine-tuning with our augmented data widen the decision boundary. Finally, we used the discriminator of the PCE as a density estimator, to identify and reject OOD data. The discriminator is trained on real/fake samples near the training distribution. Hence, we used a threshold of 0.5 on the discriminator to reject everything that is far from the training distribution. In Fig. 8.d, the white color show the input space that is rejected by the discriminator. In Fig. 8.e, we show the final uncertainty landscape without overlaying the training data. We improved the baseline model, to have high certainty only in in-distribution regions. The uncertainty increases as we go near the decision boundary. Thus in addition to image classifiers, our strategy improves the uncertainty estimates even for a classifier trained on a small 2D setup like Two Moons.

6.4. Additional Results

Much of the prior work has focused on obtaining uncertainty estimates from a pre-trained DNN output using threshold-based scoring functions. Liu et al [43] in their paper show how energy functions can be used not only as scoring functions but also as a trainable cost-function to shape the energy surface explicitly for OOD-detection. Hendrycks et al [20] propose the Outlier Exposure method which regularizes the softmax outputs to be a uniform distribution for outlier data. We compare these commonly-used methods against our technique (ACE) and show the results in Tables 3 and 4. Our method is consistent and competitive, if not outperforming, across all datasets and AiD/OOD categories.

6.5. Ablation Study

We conducted an ablation study over the three loss terms of PCE in Eq. 5. The three terms of the loss function enforces three properties of counterfactual explanation, data consistency: explanations should be realistic looking images, classifier consistency: explanations should produce a desired outcome from the classifier and self consistency: explanation image should retain the identity of the query image. For ablation study we consider the cat and dog classifier. We train three PCE, in each run we ablate one term from the final loss function. In Fig. 9, we show qualitative example of the counterfactual data augmentation generated through each PCE. Without data consistency, the images are blur and are no longer realistic. Without classifier consistency loss, though the images are realistic, but the output

Table 3. Additional results on identifying **ambiguous in-distribution (AiD)** samples. For all metrics, higher is better.

Train Dataset	Method/Model	Test-Set Accuracy	Identifying AiD	
			AUC-ROC	TNR@TPR95
AFHQ	Baseline+energy [43]	99.44±0.02	0.87±0.06	49.00±1.64
	Energy w/ fine-tune [43]	99.45±0.11	0.69±1.28	30.36±2.52
	Outlier Exposure [20]	99.50±0.14	0.85±0.01	41.07±0.75
	Baseline+ACE	99.52±0.21	0.91±0.02	50.75±3.9
Dirty MNIST	Baseline+energy [43]	95.68±0.02	0.80±0.03	17.60±0.55
	Energy w/ fine-tune [43]	96.17±0.02	0.39±0.04	11.59±0.25
	Outlier Exposure [20]	96.30±0.07	0.63±0.07	17.6±2.88
	Baseline+ACE	95.36±0.45	0.86±0.01	34.12±2.60
CelebA	Baseline+energy [43]	89.36±0.96	0.57±0.28	4.87±0.32
	Energy w/ fine-tune [43]	90.22±0.96	0.53±1.25	5.06±0.28
	Outlier Exposure [20]	86.65±1.22	0.53±0.46	5.06±0.19
	Baseline+ACE	86.80±0.79	0.74±0.06	22.36±2.30
Skin-Lesion (HAM10K)	Baseline+energy [43]	85.88±0.75	0.77±0.12	18.40±0.51
	Energy w/ fine-tune [43]	86.56±0.53	0.64±0.06	17.45±1.78
	Outlier Exposure [20]	86.37±0.46	0.73±0.02	13.21±2.70
	Baseline + ACE	81.21±1.12	0.84±0.05	71.60±3.80

Table 4. OOD detection performance for different scoring-based methods.

Train Dataset	Method	Near-OOD (Wild)		Far-OOD (CIFAR10)		Far-OOD (CelebA)	
		AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95	AUC-ROC	TNR@TPR95
AFHQ	Baseline+energy [43]	0.88±0.03	47.77±1.10	0.94±0.05	72.68±2.69	0.96±0.04	74.75±2.89
	Energy w/ fine-tune [43]	0.93±3.06	45.97±2.78	0.99±0.00	0.66±0.01	0.94±1.86	68.38±3.03
	Outlier Exposure [20]	0.92±0.01	73.99±2.62	0.99±0.20	99.54±0.79	0.96±0.01	78.69±3.02
	Baseline+ACE	0.89±0.03	51.39±4.40	0.98±0.02	88.71±5.70	0.97±0.03	88.87±9.80
Dirty MNIST	Baseline+energy [43]	0.87±0.04	40.30±1.05	0.86±0.12	43.92±2.30	0.91±0.02	62.10±5.17
	Energy w/ fine-tune [43]	0.60±0.08	37.43±0.93	1.00±0.00	99.99±0.00	1.00±0.00	99.06±0.01
	Outlier Exposure [20]	0.94±0.01	65.58±1.64	1.00±0.00	99.99±0.00	1.00±0.00	99.56±0.12
	Baseline+ACE	0.94±0.02	37.23±1.90	0.98±0.02	67.88±3.10	0.97±0.02	70.71±1.10
CelebA	Baseline+energy [43]	0.76±0.51	9.40±0.01	0.94±0.08	32.08±1.70	0.85±0.76	17.10±0.72
	Energy w/ fine-tune [43]	0.85±1.27	32.81±1.92	0.99±0.00	99.99±0.00	0.91±0.77	84.35±1.29
	Outlier Exposure [20]	0.66±0.69	8.44±0.45	0.75±0.70	26.09±0.51	0.69±0.53	16.63±0.90
	Baseline+ACE	0.87±0.03	34.37±2.50	0.96±0.01	96.35±2.50	0.92±0.05	63.51±1.50
Skin-Lesion (HAM10K)	Baseline+energy [43]	0.70±0.04	10.85±0.08	0.70±0.14	7.90±0.29	0.65±0.20	2.83±1.33
	Energy w/ fine-tune [43]	0.62±0.02	9.80±1.81	1.00±0.00	99.77±0.33	0.76±0.13	16.04±1.08
	Outlier Exposure [20]	0.67±0.09	10.38±3.30	0.99±0.00	97.17±2.37	0.81±0.08	22.64±4.30
	Baseline+ACE	0.72±0.04	10.99±2.80	0.97±0.02	66.77±1.40	0.96±0.03	95.83±5.00

of the classifier is not changing with the condition, hence such PCE won't generate augmented samples near the decision boundary, which is the goal of our proposed strategy. With self consistency, the generated images are not a gradual transformation of a given query image.

Further, in Fig. 10 we present quantitatively compare the uncertainty estimates from the baseline, before and after the fine-tuning with ACE. In each row, we represent a different ablation over the three loss terms. Fig. 10.A. shows the predicted entropy (PE) of **in-distribution (ID)** samples. Ideally, fine-tuning should minimally effect the PE distribution over iD samples. Without classification consistency loss (second row), the PE distribution of iD samples changed significantly. Fig. 10.B and Fig. 10.C shows the PE distribution over **ambiguous in-distribution (AiD)** samples and

near-OOD samples, respectively. The data augmentation derived from PCE without adversarial loss or reconstruction loss, is not able to separate AiD samples or near-OOD from rest of the test set. In Fig. 10.D, we use the discriminator of the PCE to identify **far-OOD** samples. In all three rows, we observe sub-optimal performance of the discriminator in identifying and rejecting far-OOD samples. The legend shows the AUC-ROC for binary classification over uncertain samples and iD samples. Hence, all three loss terms are important to improve the uncertainty estimates of the baseline over all samples across the uncertainty spectrum.

6.6. Robust Generalization

In this experiment, we establish a connection between loss landscape plots and generalization of classifiers. In or-

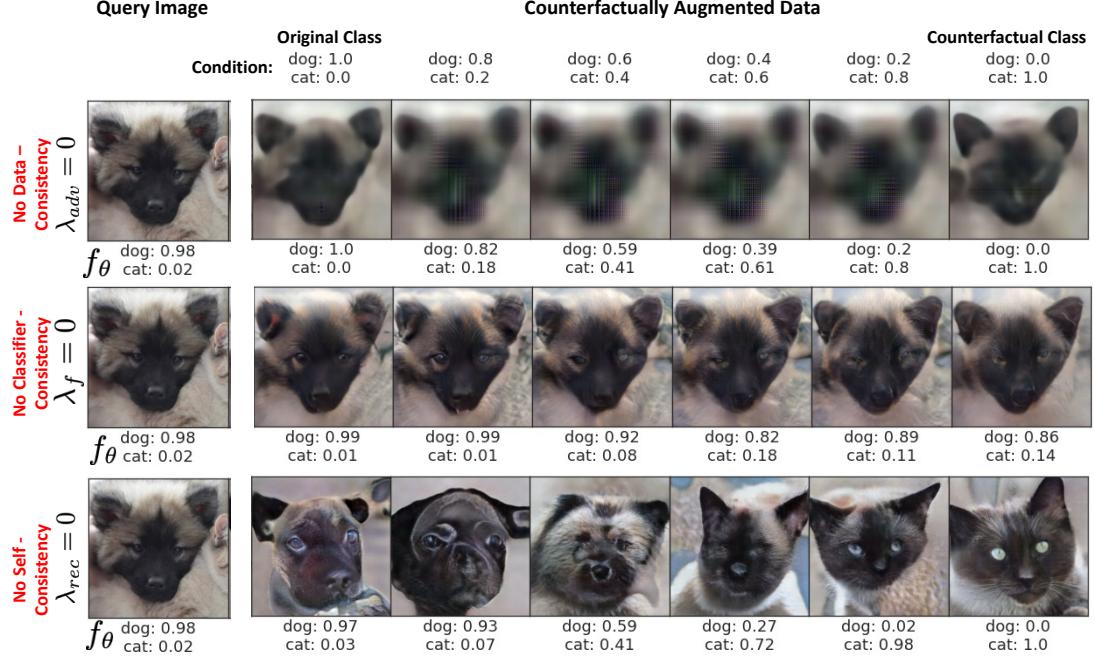


Figure 9. Examples of data augmentation while ablating different loss terms.

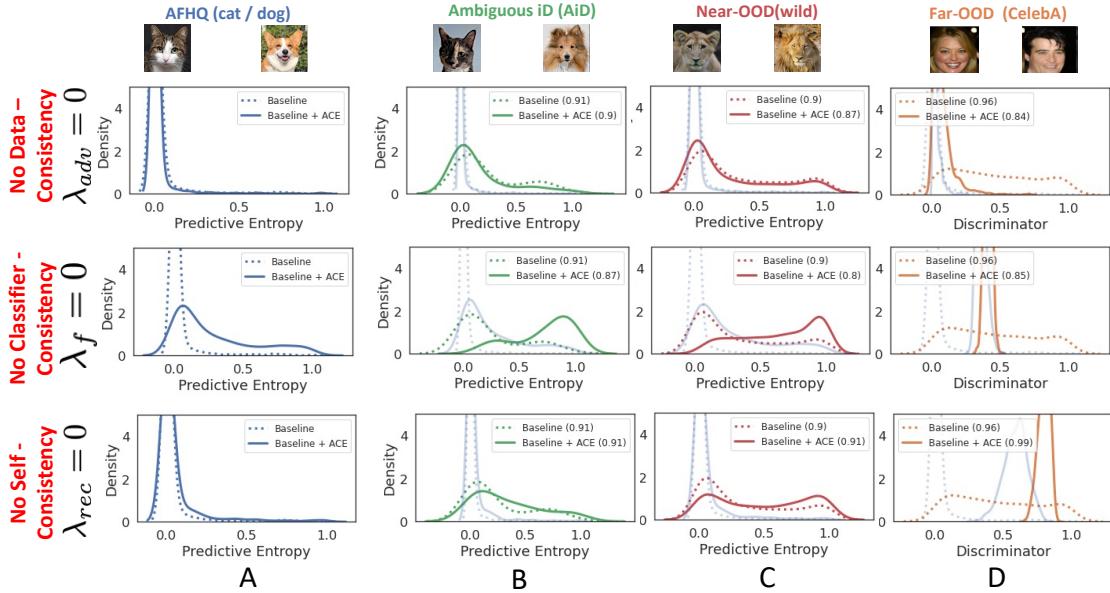


Figure 10. Comparison of the uncertainty estimates from the baseline, before and after the fine-tuning with ACE. Each row represents a different ablation over the three loss terms. A) Predicted entropy (PE) of **in-distribution (iD)** samples. Ideally, fine-tuning should minimally effect the PE distribution over iD samples. Without classification consistency loss (second row), the PE distribution of iD samples changed significantly. B) PE distribution over **ambiguous in-distribution (AiD)** samples. C) PE distribution over **near-OOD** samples. The data augmentation derived from PCE without adversarial loss or reconstruction loss, is not able to separate AiD samples or near-OOD from rest of the test set. D) We use the discriminator of the PCE to identify **far-OOD** samples. In all three rows, we observe sub-optimal performance of the discriminator in identifying and rejecting far-OOD samples. The legend shows the AUC-ROC for binary classification over uncertain samples and iD samples. Hence, all three loss terms are important to improve the uncertainty estimates of the baseline over all samples across the uncertainty spectrum.

der to qualitatively understand the improved generalization of our method, we try to visualize the high-dimensional loss landscape via 3D weight visualization plots as shown by Li *et al.* We compute the cross-entropy loss using test set of CelebA and AFHQ and follow the method given by Li *et al.* to compare the loss landscape geometry for the baseline model and our method (ACE).

We observe that our method leads to smooth and flatter loss landscapes as compared to baseline. This shows that slight perturbation to the weight does not change the loss much, which may qualitatively explain why we obtain better generalization performance and robustness to adversarial attacks in our experiments. We do not thoroughly investigate this direction and leave it as an important direction for future research.

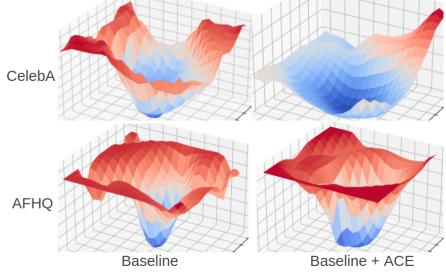


Figure 11. Weight loss landscape visualizations for baseline model and our method on CelebA and AFHQ datasets