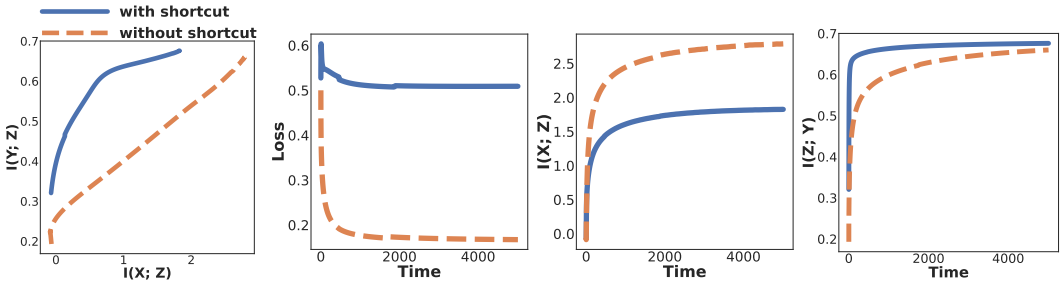
A APPENDIX

A.1 ADDITIONAL EXPERIMENT

CelebA with synthetic shortcut. We also validate our hypothesis on CelebA dataset (Liu et al., 2015). Similar to the MNIST experiment, we created a synthetic shortcut by adding a small white patch on one corner of training images tagged as *male*. The model is trained for the binary classification of images into CelebA's male and female classes. As shown in Figure 9, the mutual information $I(X;Z)$ on the OOD test data converges to a lower value in the presence of shortcuts in the training data. It can be observed from Figure 9a that the model can achieve higher accuracy on training data even by encoding less information about the input space using shortcuts present in the training data.



(a) Information Plane (b) Loss on test data (c) $I(X;Z)$ (d) $I(Z;Y)$

Figure 9: Mutual information profile for the CelebA dataset with synthetic shortcuts. Plot of $I(X;Z)$ in (a) and (c) show that shortcuts results in reduced $I(X;Z)$. Animated GIF of the plot can be viewed here.

A.2 IMPLEMENTATION DETAILS

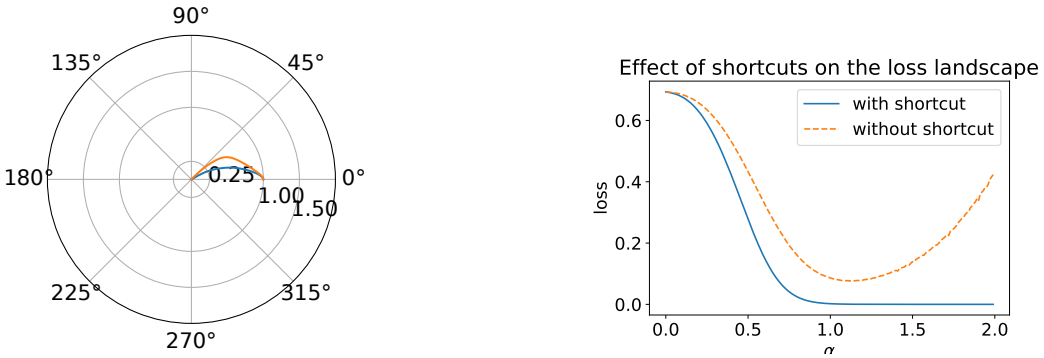
We used the Neural Tangent library (Novak et al., 2019) to compute the NTK and NNGP kernel for a given architecture; JAX (Bradbury et al., 2018) to implement the infinite-width neural network and compute mutual information. We used publicly available datasets for our experiments and sampled the dataset to introduce spurious correlation with the class labels. Code for implementing our method along with the datasets can be found on our **anonymous repository**.

	MNIST		**CelebA**		**Waterbird**		**NICO**	
	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR
Energy	0.27	0.99	0.47	0.91	0.51	0.90	0.73	0.84
Entropy	0.50	1.00	0.50	1.00	0.50	1.00	0.75	0.74
ODIN	0.50	1.00	0.50	1.00	0.50	1.00	0.73	0.79
Mahalanobis	0.69	0.51	0.85	0.80	0.64	0.94	0.67	0.67
MaxLogit	0.26	0.99	0.47	0.91	0.51	0.90	0.75	0.81
MaxSoftmax	0.50	1.00	0.50	1.00	0.50	1.00	0.76	0.75
MCD	0.50	1.00	0.50	1.00	0.50	1.00	0.50	1.00

Table 2: AUROC and FPR@95TPR (denoted by FPR in the table) values of OOD detectors on different datasets. While Mahalanobis can detect shortcuts in MNIST, CelebA and NICO, it fails to detect shortcuts in the waterbird dataset. We used $\tau = 0.90$ to threshold FPR values.

A.3 BASELINES

We benchmark against the following OOD baselines to show our method can detect shortcuts while the existing OOD detectors cannot:



(a) Radial plot

(b) Loss landscape visualization

Figure 10: Visualization of loss landscape on MNIST dataset. (a). polar coordinates (r_t, ϕ_t) measuring the deviation from the linear path between initialisation and converged parameters in the weight space during the optimization. (b). 1-D visualization of loss landscape.

Energy-based OOD detector. Liu et al. (2021b) proposed an energy-based method to detect OOD inputs using an energy score. The model maps the input to a single, non-probabilistic scalar called the energy. The method uses energy instead of softmax for calculating the confidence scores. Data samples with high energy are considered as OOD inputs and vice versa.

Entropy-based OOD detector. Macedo et al. (2022) introduced IsoMax loss to train the model, which improves the OOD detection to tackle the overconfidence of SoftMax loss. IsoMax loss force the logits to depend only on the distances from the high-level features to the class prototypes. Let $f_\theta(x)$ represent the feature embeddings for the input x, p_θ^j represent the learnable prototype with class j, and y^k represent the label of the correct class; IsoMax loss can be described as following:

$$L_I(y^k|x) = -\log \frac{\exp(-d(f_{\theta(x)}, p_\theta^k))}{\sum_j \exp(-d(f_{\theta(x)}, p_\theta^j))} \tag{10}$$

Monte Carlo Dropout (MCD). Gal & Ghahramani (2016) introduced MCD, which uses Dropout (Srivastava et al., 2014) as an Bayesian approximation to the Gaussian Processes. MCD uses variance of the output probabilistic distribution to estimate the model's confidence and detect OOD samples.

Mahalanobis. Lee et al. (2018b) proposed to measure the probability density of test samples in feature spaces using class-conditional Gaussian distribution. They defined the confidence score using *Mahalanobis* distance with respect to the closest class-conditional distribution, where its parameters are chosen as empirical class means and tied to the empirical covariance of training samples.

MaxSoftmax. Hendrycks & Gimpel (2018) observed that correctly classified examples tend to have greater maximum softmax probabilities than incorrectly classified and OOD. They showed that the prediction probability of OOD samples is lower than the prediction probability of in-distribution samples, and thus, observing prediction probability statistics can help in detecting OOD samples.

MaxLogit. Hendrycks et al. (2022) proposed to use the negative of the maximum unnormalized logit for an anomaly score $-\max_k f(x)_k$, which they call *MaxLogit* as a confidence score for detecting OOD samples.

ODIN. Liang et al. (2020) proposed a simple change to softmax to improve OOD detection. ODIN used a temperature scaling in the softmax and adds small perturbations to the training inputs for more effective OOD detection.

A.4 EFFECT OF SHORTCUT ON THE LOSS LANDSCAPE:

We visualize the loss landscape of neural networks to understand the effect of shortcuts on the optimization trajectory. We plot loss along a linear path connecting the initial parameter θ_o and converged parameter θ^* in the weight space (Goodfellow et al., 2014) and polar coordinates (r_t,ϕ_t) plot measuring the deviation from the linear line between θ_i and θ^* (Figure 10). We parameterize the line with α such that $\theta = (1-\alpha)\theta_i + \alpha\theta^*$. Polar coordinates can be calculated using $r_t = \frac{||\triangle\theta_t||}{||\triangle\theta_o||}$ and $\phi_t = \arccos\frac{\triangle\theta_t \times \triangle\theta_o}{||\triangle\theta_t|| \times ||\triangle\theta_o||}$, where $\triangle\theta_t = \theta_t - \theta^*$. We observe that the loss landscape around θ^* in the case of shortcuts is surprisingly flat as compared to the valley-like shape for a model trained on data not containing shortcuts using the MNIST dataset. The polar plot shows that the optimizer deviates less from the linear trajectory when trained with shortcuts.

A.5 ADDITIONAL COMMENTS

We chose to use NTK to calculate mutual information, as in contrast to other methods such as MINE (Belghazi et al., 2021), NTK doesn't require training using gradient descent due to its kernel behaviour at the infinite limit. NTK can give the mutual information profile during the entire training evolution without much computation overhead, whereas MINE and other neural network-based methods need to be trained for hundreds of epochs to approximate the MI between the two variables, which would be feasible and computationally expensive to calculate MI for every epoch during the training evolution. To summarise, our work is different from existing shortcut detection methods in the literature as the proposed method doesn't require any human annotation or human-in-the-loop to detect shortcuts, i.e., our method is domain agnostic and does not require human expertise to detect shortcuts. Moreover, due to the kernel behaviour of NTK, mutual information can be computed for the entire training evolution without much computational overhead. Our method can be used to check for shortcuts in the training data before deploying the model on new unlabelled test data. For e.g., in the medical dataset experiment, we trained the model on the Messidor dataset and used APTOS dataset as a test dataset. Our method was able to detect shortcuts in the context of the APTOS and Messidor dataset (Figure 9), which is in line with the recent benchmarking result on these two datasets, i.e., models trained on APTOS do not generalize well [3]. We believe this is quite a useful application of our method, especially in domains where it is difficult to detect shortcuts manually.