A Causal View on Robustness of Neural Networks

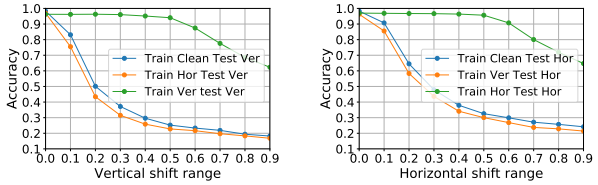
Cheng Zhang[*1] **Kun Zhang**[2] **Yingzhen Li**[*1]

Abstract

We present a causal view on the robustness of neural networks against input manipulations, which applies not only to traditional classification tasks but also to general measurement data. Based on this view, we design a deep causal manipulation augmented model (deep CAMA) which explicitly models possible manipulations on certain causes leading to changes in the observed effect. We further develop data augmentation and test-time fine-tuning methods to improve deep CAMA's robustness. When compared with discriminative deep neural networks, our proposed model shows superior robustness against unseen manipulations. As a by-product, our model achieves disentangled representation which separates the representation of manipulations from those of other latent causes.

1. Introduction

Deep neural networks (DNNs) have great success in many real-life applications, however, they are easily fooled even by a tiny amount of perturbation (Szegedy et al., 2013; Goodfellow et al., 2015; Carlini & Wagner, 2017b; Athalye et al., 2018). Lack of robustness hinders the application of DNNs to critical decision making tasks such as uses in health care. To address this, a deep learning practitioner may suggest training DNNs with datasets that are not only big but also diverse. Indeed, data augmentation and adversarial training have shown improvements in both the generalization and robustness of DNNs (Kurakin et al., 2016; Perez & Wang, 2017; Madry et al., 2017). Unfortunately, this does not address the vulnerability of DNNs for unseen manipulations. For example, as shown in Figure 1, a DNN trained on clean MNIST digits fails to classify shifted digits. Although observing (adversarial) perturbations of clean data in training improves robustness against that particular manipulation (the green line), the DNN is still fragile when unseen manipulations are present (orange line). Since it is unrealistic to augment the training data towards all possible manipulations that many occur, a principled method that fundamentally improves the robustness is much needed.



(a) Test Vertical shift (b) Test Horizontal shift

Figure 1. Robustness results for DNNs on shifted MNIST. Panels (a) and (b) show the accuracy on classifying noisy test data generated by shifting the digits vertically (Ver) and horizontally (Hor). It shows that data augmentation during training makes generalization to unseen shifts worse (orange versus blue lines).

On the other hand, thanks to the generative view, or the capability of causal reasoning, human perception is adaptive and robust to such perturbations. After learning the concept of an "elephant", a child can identify the elephant in a photo taken under any lightning condition, location, etc. In the causal view, the lightning condition and the location are causes of the presented scene, which can be intervened without changing the presence of the elephant. However, many machine learning methods do not take possible interventions into account, but make use of only the provided data, and cannot adapt the predictor for new data gathered with unseen manipulation. Therefore we argue that the incapability of *causal reasoning* (Pearl & Mackenzie, 2018; Gopnik et al., 2004) is the reason of DNN's vulnerability to (adversarial) data manipulations.

This work discusses the robustness of DNNs from a causal perspective. Our contributions are:

- *A causal view on robustness of neural networks.* We argue from a causal perspective that adversarial examples for a model can be generated by manipulations on the effect variables and/or their unseen causes. Therefore DNN's vulnerability to adversarial attacks is due to the lack of causal understanding.
- *A causal inspired deep generative model.* We design a causal inspired deep generative model which takes into account possible interventions on the causes in the data generation process (Woodward, 2005). Accompanied with this model is a test-time in-

1. Microsoft Research 2. Carnegie Mellon University; * equal contribution; Correspondence to: Cheng Zhang <cheng.zhang@microsoft.com>

ference method to learn unseen interventions and thus improve classification accuracy on manipulated inputs. Compared to DNNs, experiments on both MNIST and a measurement-based dataset show that our model is significantly more robustness to unseen manipulations.

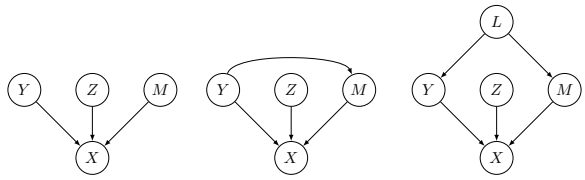
2. A Causal View on Robustness of Neural Networks

Discriminative DNNs are not robust to manipulations such as adversarial noise injection (Goodfellow et al., 2015; Carlini & Wagner, 2017a; Athalye et al., 2018), rotation and shift. They simply trust the observed data and ignore the constraints of the data generating process, which leads to overfiting to nuisance factors that do not cause the ground truth labels. By exploiting the overfit to the nuisance factors, an adversary can easily manipulate the inputs to fool discriminative DNNs into predicting the wrong outcomes.

On the contrary, human can easily recognize an object in a scene and be indifferent to the variations in other aspects such as background, viewing angle, the presence of a sticker to the object, etc. More importantly, human recognition is less affected even by drastic perturbations, e.g. variations in the lighting condition. We argue that the main difference here is due to our ability to perform *causal reasoning*, which identifies factor that are not relevant to the recognition results (Freeman, 1994; Peters et al., 2017; Parascandolo et al., 2017). This leads to robust human perception to not only a certain type of perturbations, but also to many types of unseen manipulations which is caused by intervention on other factors. Thus we argue that one should incorporate causal perspective into model design, and make the model robust on the level of different types of manipulations.

Before presenting our causally informed model, we first define the generative process of perceived data. There might exist multiple causes in the data generation process influencing the observed data X, and we visualize exemplar causal graphs in Figure 2 with the arrows indicating causal associations. Among these causes of X, Y is the target to be predicted, M is a set of variables which can be intervened artificially, and Z represents the rest of the causes that cannot be intervened in the application context. Take hand-written digit classification for example, X is the image and Y is the class label. The appearance of X is an effect of the digit number Y, latent causes Z such as writing styles, and possible manipulations M, such as rotation or translation.

We can thus define valid attacks through the lens of causality. Datasets are produced by interventions in general, so do adversarial examples. Therefore defining a valid attack is equivalent to defining a set of variables in the causal graph (Figure 2) which can be intervened by the adversary. We argue that *a valid attack is an intervention on M which, together with the original Y and Z, produces the manipulated data X*. We do not consider interventions on Y and Z (and their causes): interventions on Y (and its causes) changes the "true" value of the target and do not correspond to the type of attacks we are considering; by definition Z (and its causes) cannot be intervened *artificially* thereby unavailable to the adversary. In this regard, recent adversarial attacks can be considered as a specific type of intervention on M such that a learned predictor is deceived.



(a) factorized (b) Y causes M (c) confounded

Figure 2. Exampler causal graphs with Y, Z, M causing X. Y might cause M (panel b), or they might be confounded (panel c).

In light of the above definition on valid attacks, it is clear that performing prediction adaptive to the (unknown) intervention is necessary to achieve robustness to manipulated data. A natural way to build such adaptive predictor is to construct a model that perform reasoning in a way consistent to the causal process. To see this, note that a valid attack changes the value of M, but it leaves the functional relationship from M and Y to X intact. This is known as *modularity property* (Woodward, 2005), and in this sense the causal system is autonomous (Pearl, 2009). Therefore a causally consistent predictive model is expected to be able to learn this functional relationship from data, and adapt the prediction result of target in test time according to its reasoning on the underlying causal factors.

3. The Causal Manipulation Augmented Model

We propose a deep CAusal Manipulation Augmented model (deep CAMA), which takes into account the causal relationship for model design. We also design a fine-tuning algorithm to enable adaptive reasoning of deep CAMA for unseen manipulations on effect variables. The robustness can be further improved by training-time data augmentation, without sacrificing the generalization ability to unseen manipulations. Below we first present the deep CAMA for single modality data, and then present a generic deep CAMA for multimodality measurement data.

3.1. Deep CAMA for single modality data

The task of predicting Y from X covers a wide range of applications such as image/speech recognition and sentiment analysis. Normally a discriminative DNN takes X as input and directly predicts (the distribution of) the target variable

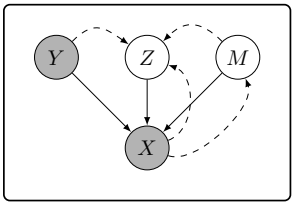


Figure 3. Graphical presentation of proposed causally consistent deep generative model for single modal data.

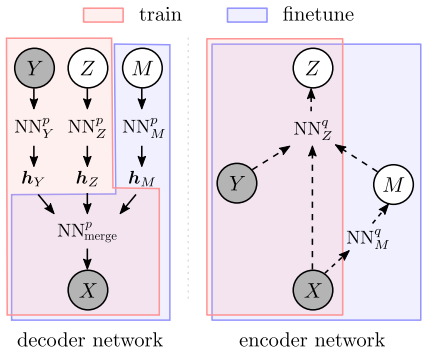


Figure 4. The network architecture. Shaded areas show the selective part for $do(m)$ training and the fine-tune method, respectively.

Y. Generative classifiers, on the other hand, build a generative model $Y \to X$, and use Bayes' rule for predicting Y given X: $p(y|x) = p(y)p(x|y)/p(x)$.

We design deep CAMA (Figure 3) following the causal graph in Figure 2(a), which returns a factorized model:

$$p_\theta(x, y, z, m) = p(m)p(z)p(y)p_\theta(x|y, z, m). \quad (1)$$

Notice that we do not consider modelling dependencies between Y and M even when the causal relationship might exist (see Figures 2(b) and 2(c)) in the generation process of the training data. By our definition of valid attack, M is intervened (i.e. $do(m)$), which blocks the influence from Y to M, and the generation process of manipulated data reduces to the factorised case (Figure 2(a)).

For efficient inference we use *amortized inference* (Kingma & Welling, 2013; Rezende et al., 2014; Zhang et al., 2018) and define an inference network for posterior approximation:

$$q_\phi(z, m|x, y) = q_{\phi_1}(z|x, y, m)q_{\phi_2}(m|x). \quad (2)$$

We use ϕ to denote all the parameters of the encoder network; here $\phi = \{\phi_1, \phi_2\}$, where ϕ_1 is the parameter for the encoder network for the variational distribution $q_{\phi_1}(z|x, y, m)$, and ϕ_2 is used for the $q_{\phi_2}(m|x)$ part. Here we assume that given X, Y does not contain further information about M (as a consequence, Y and M are conditionally independent given X, although it is not implied in the graphical representation), and it is clearly the case if X contains all the information of M. Therefore in $q_{\phi_2}(m|x)$ we only extract the information of M form X, which, as we shall show later, allows deep CAMA to learn unseen manipulations.

The network architecture is presented in Figure 4. For the p model, the cause variables Y, Z and M are first transformed into feature vectors h_Y, h_Z and h_M. Later, these features are merged together and then passed through another neural network to produce the distributional parameters of $p_\theta(x|y, z, m)$. For the approximate posterior q, two different networks are used to compute the distributional parameters of $q_{\phi_2}(m|x)$ and $q_{\phi_1}(z|x, y, m)$, respectively.

Model training We describe the training procedure for two different scenarios. First, assume that during training, the model observes clean data $\mathcal{D} = \{(x_n, y_n)\}$ only. In this case we set the manipulation variable M to a null value, e.g. $do(m = 0)$. and train deep CAMA by maximizing the likelihood function $\log p(x, y|do(m = 0))$ under training data. As there is no incoming edges to the manipulation variable M, the do-calculus can be reduced to the conditional distribution $p(x, y|do(m = 0)) = p(x, y|m = 0)$. Since this marginal distribution is intractable, we instead maximize the intervention evidence lower-bound (ELBO) with $do(m = 0)$, i.e. $\max_{\theta,\phi} \mathbb{E}_{\mathcal{D}}[\text{ELBO}(x, y, do(m = 0))]$, with the ELBO derived in appenfix A and defined as:

$$\begin{aligned}&\text{ELBO}(x, y, do(m = 0))\\ :=&\mathbb{E}_{q_{\phi_1}(z|x,y,m=0)}\left[\log \frac{p_\theta(x|y, z, m = 0)p(y)p(z)}{q_{\phi_1}(z|x, y, m = 0)}\right].\end{aligned} \quad (3)$$

If manipulated data $\mathcal{D}'$ is available during training, then similar to data augmentation and adversarial training (Goodfellow et al., 2015; Tramèr et al., 2018; Madry et al., 2017), we can augment the training data with such data. We still use the intervention ELBO (3) for clean data. For the manipulated instances, we can either use the intervention ELBO with $do(m = m_0)$ when the manipulated data $\mathcal{D}' = \{(m_0(x), y)\}$ is generated by a known intervention m_0, or, as done in our experiments, infer the latent variable M for unknown manipulations. This is achieved by maximizing the ELBO on the joint distribution $\log p(x, y)$ using manipulated data:

$$\text{ELBO}(x, y) := \mathbb{E}_{q_\phi(z,m|x,y)}\left[\log \frac{p_\theta(x, y, z, m)}{q_\phi(z, m|x, y)}\right], \quad (4)$$

so the total loss function to be maximized is defined as

$$\begin{aligned}\mathcal{L}_{\text{aug}}(\theta, \phi) =&\lambda\mathbb{E}_{\mathcal{D}}[\text{ELBO}(x, y, do(m = 0))]\\ &+ (1 - \lambda)\mathbb{E}_{\mathcal{D}'}[\text{ELBO}(x, y)].\end{aligned} \quad (5)$$

Our causally consistent model effectively disentangles the latent representation: Z models the unknown causes in the clean data, such as personal writing style; and M models possible manipulations or intervention on the underlying

factors, which the model should be robust to, such as shift, rotation, noise etc. From a causal perspective, the mechanism of generating X from its causes is invariant to the interventions on M. Thus, in our model the functional relationships $Y \to X$ and $Z \to X$ remain intact even in the presence of manipulated data. As a result, deep CAMA's can still generalize to unseen manipulations even after seeing lots of manipulated datapoints from other manipulations, in contrast to the behavior of discriminative DNNs as shown in Figure 1.

Prediction In general the test data $\tilde{\mathcal{D}}$ can be manipulated with an unseen intervention, for which we wish our model to be robust to. Thus, at test-time, M is unknown, and deep CAMA classifies an unseen test data x^*, using a Monte Carlo approximation to Bayes' rule with samples $m^u \sim q_{\phi_2}(m|x)$, $z_c^k \sim q_{\phi_1}(z|x^*, y_c, m^u)$:

$$p(y^*|x^*) = \frac{p(x^*|y^*)p(y^*)}{p(x^*)} \tag{6}$$
$$\approx \text{softmax}_{c=1}^{C}\left[\log \sum_{k=1}^{K} \frac{p_\theta(x|y, z_c^k, m^u)p(y_c)p(z)}{q_{\phi_1}(z_c^k|x^*, y_c, m^u)}\right].$$

In addition, deep CAMA can be adapted to the unseen manipulations present at test time *without labels on the manipulated data*. From the causal graph, the conditional distributions $p(x|y)$ and $p(x|z)$ are invariant to the interventions on X based on the modularity property. However, we would like to learn the manipulation mechanism $M \to X$, and, given that the number of possible interventions on M might be infinity, the model may be underfitted for this functional relationship, given limited data. Fine-tuning on the current observation can be beneficial to address this underfiting issue, thereby hopefully making deep CAMA more robust. As shown in Figure 4, for the generative model, we only fine-tune the networks that are dependent only on M, i.e. NN_M^p by maximizing the ELBO of the marginal distribution $\log p(x)$:

$$\text{ELBO}(x) := \log\left[\sum_{c=1}^{C} \exp[\text{ELBO}(x, y_c)]\right]. \tag{7}$$

To reduce the possibly negative effect of fine-tuning to model generalization, we use a shallow network for NN_{merge}^p and deep networks for NN_M^p, NN_Y^p and NN_Z^p. We also fine-tune the network NN_M^q for the approximate posterior q since M is involved in the inference of Z. In sum, in fine-tuning the selective part of the deep CAMA model is trained to maximize the following objective:

$$\mathcal{L}_{\text{ft}}(\theta, \phi) = \alpha\mathbb{E}_{\mathcal{D}}[\text{ELBO}(x, y)] + (1-\alpha)\mathbb{E}_{\tilde{\mathcal{D}}}[\text{ELBO}(x)]. \tag{8}$$

The intervention ELBO can also be used for $\mathcal{D}$.

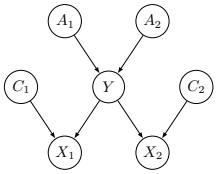


Figure 5. The Markov Blanket of target variable Y

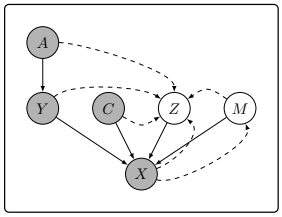


Figure 6. Graphical presentation of deep CAMA for generic measurement data.

Notice that there may exist infinitely many manipulations and it is impossible to observe all of them at training time. Therefore by fine-tuning at test-time, the model can be adapted to unseen manipulation which is desirable in many real-life applications. As shown in our experiments, the proposed deep CAMA model and the training methods are capable of improving the robustness of the generative classifier to unseen manipulations.

3.2. Deep CAMA for generic measurement data

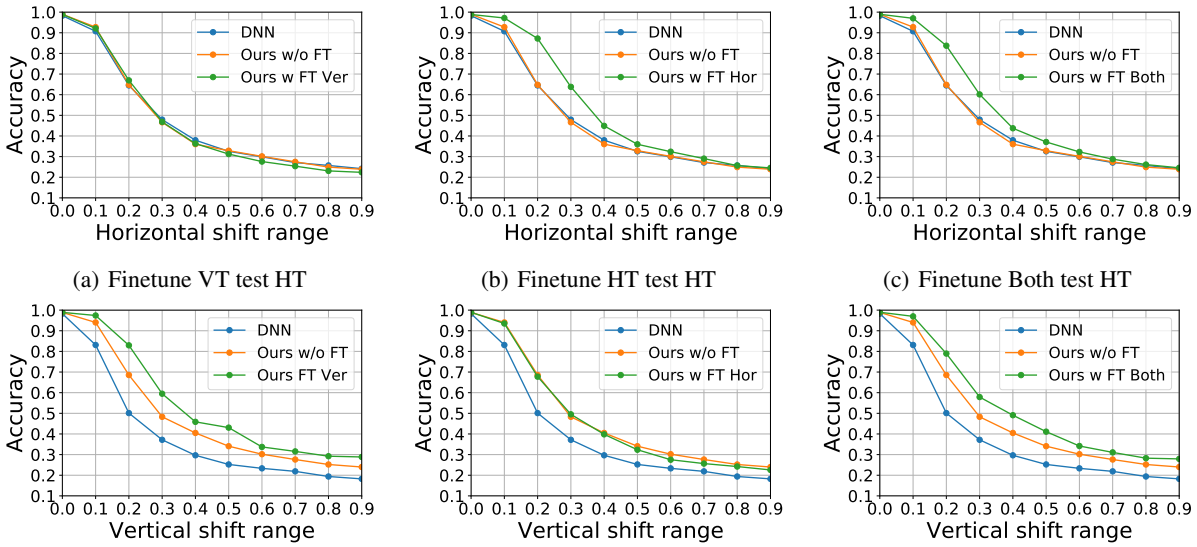
We now discuss a generic version of deep CAMA to handle multimodality in measurement data. To predict the target variable Y in a directed acyclic graph, only variables in the Markov blanket of Y (shown in Figure 5) are needed. This includes the parents (A), children (X), and co-parents (C) of the target Y. Similar to the single modal case above, here a valid manipulation can only be independent mechanisms applied to X or C to ensure that both Y and the relationship from Y to X remain intact.

We design the generic deep CAMA (shown in Figure 6) following the causal process in Figure 5. Unlike discriminative DNNs where A, C and X are used together to predict Y directly, we consider the full causal process and treat them separately. Building on the deep CAMA for single modality data, we add the extra consideration of the parent and observed co-parent of Y, while modelling the latent unobserved cause in Z and potential manipulations in M. We do not need to model manipulation on C as they are out of the Markov Blanket of Y. Thus, our model and the approximate inference network are defined as:

$$\begin{aligned} &p_\theta(x, y, z, m, a, c) \\ &= p(a)p(m)p(z)p(c)p_{\theta_1}(y|a)p_{\theta_2}(x|y, c, z, m), \end{aligned} \tag{9}$$

$$q_\phi(z, m|x, y, a, c) = q_{\phi_1}(z|x, y, m, a, c)q_{\phi_2}(m|x). \tag{10}$$

Training, fine-tuning and prediction proceed in the same way as in the single modality case (Section 3.1) with $do(m)$ operations and Monte Carlo approximations. As we only fine-tune the networks that are dependent on M, similar reasoning indicates that the multimodality deep CAMA is robust to manipulations directly on the effect variable X.



(a) Finetune VT test HT (b) Finetune HT test HT (c) Finetune Both test HT

(d) Finetune VT test VT (e) Finetune HT test VT (f) Finetune Both test VT

Figure 7. Model robustness results on horizontal shifts (top) and vertical shifts (bottom).

Our model is also robust to changes of X caused by intervention on the co-parents C by design. By our definition of valid manipulation, perturbing C is valid as it only leads to the changes in X. If the underlying model from Y and C to X remains the same, and the trained model learns $p(x|y,c)$ *perfectly*, then our model is perfectly robust to such changes. This is because we use Bayes' rule for prediction:

$$\begin{aligned} p(y|a,x,c) =& \frac{p(y|a)p(a)p(c)p(x|y,c)}{p(a)p(c)\int_y p(y|a)p(x|y,c)} \\ =& \frac{p(y|a)p(x|y,c)}{\int_y p(y|a)p(x|y,c)}, \end{aligned} \tag{11}$$

and the manipulations on C (thus changing X) do not affect the conditional distribution $p(x|y,c)$ in the generative classifier (Eq. 11). In contrast, discriminative DNNs concatenate X, C, A together and map these variables to Y, and therefore it fails to make use of the invariant mechanisms.

4. Experiments

In this section, we first show the robustness of deep CAMA for image classification using both MNIST and a binary classification task derived from CIFAR-10. Then, we demonstrate the behaviour of our generic deep CAMA for measurement data. We evaluated the performance of CAMA on both manipulations such as shifting and adverserial examples generated using the CleverHans package (Papernot et al., 2018). More results with different DNN architectures and different manipulations are shown in the appendix.

4.1. Robustness test on image classification with Deep CAMA

We first demonstrate the robustness of our model against vertical (VT) and horizontal (HT) shifts. Details such as network architectures are presented in the appendix.

Training with clean MNIST data only. Figure 7 shows the results for deep CAMA trained on clean data only. Deep CAMA without fine-tuning (orange lines) perform similarly to a DNN (blue lines) on horizontally shifted images, but it is more robust to vertical shifts. The advantage of deep CAMA is clear when fine-tuning is used at test time (green lines): fine-tuning on manipulated test data with the same shift clearly improves the robustness of the network (panels 7(b) and 7(d)). We further inspect the generalization of deep CAMA to unseen manipulation after fine-tuning in unrelated manipulation in panels 7(a) and 7(e). The robustness of the model fine-tuned for other manipulated data does not drop, which is desired and different to discriminate DNN. This shows that our model is capable of learning manipulations in an unsupervised manner, without deteriorating the generalization ability to unseen manipulations. Lastly, panels 7(c) and 7(f) show the robustness of our model to both shifts when both types of manipulation are used for fine-tuning, and we see clear improvements over both manipulations.

Training with augmented MNIST data We explore the setting where the training data is augmented with manipulated data. As discussed in Section 3.1, here deep CAMA naturally learns disentangled representation due to its causal reasoning capacity. Indeed this is confirmed by Figure 9,

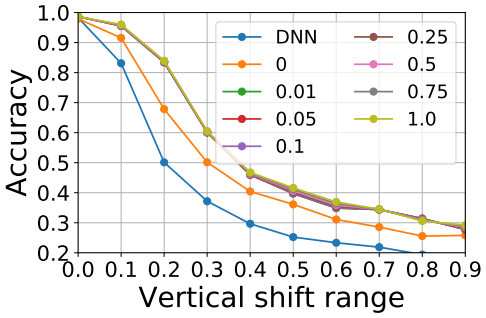
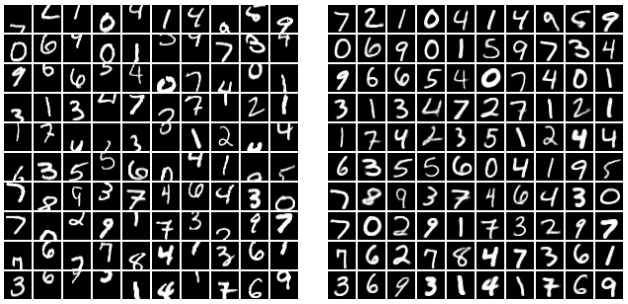
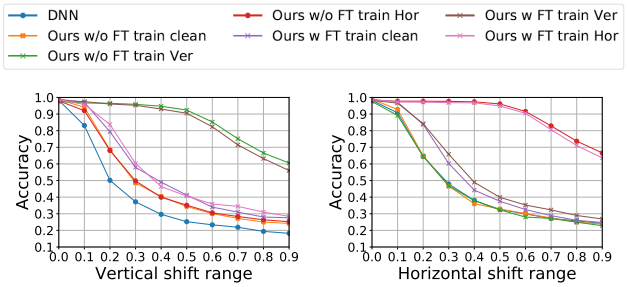


Figure 8. Performance regarding different percentages of test data used for fine-tuning manipulation



(a) Vertically shifted data (b) do(m=0) with the z and y from the vertical shifted data

Figure 9. Visualization of the disentangled representation.



(a) Test Vertical shift (b) Test horizontal shift

Figure 10. Performance of our model against different manipulation (c.f. Figure 1).

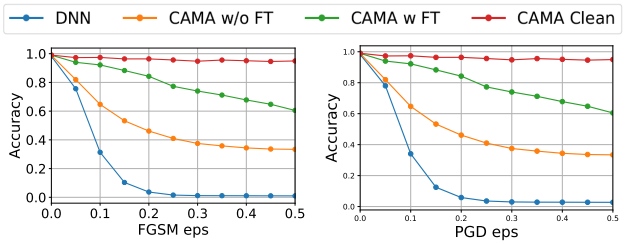


Figure 11. Test accuracy on MNIST adversarial examples.

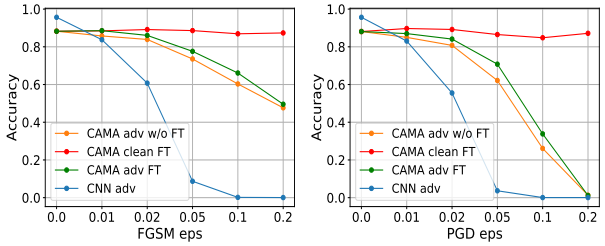
where panel 9(b) shows the reconstructions of manipulated data from panel 9(a) with $do(m = 0)$. In this case the model keeps the identity of the digits but moves them to the center of the image. Recall that $do(m = 0)$ corresponds to clean data which contains centered digits. This shows that deep CAMA can disentangle the intrinsic unknown style Z and the shifting manipulation variable M.

We show the robustness results of deep CAMA with augmented training in Figure 10 (cf. Figure 1). Here shift range 0.5 is used to augment the training data. Take the vertical shift test in panel 10(a) for example. When vertically shifted data are augmented to the training set, the test performance without fine-tuning (green line) is significantly better. Further, fine-tuning (brown line) brings in even larger improvement for large scale shifts. On the other hand, deep CAMA maintains robustness on vertically shifted data when trained with horizontally shifted data, which is different from discriminative DNN's overfiting behaviour (Figure 1). Therefore, our model does not overfit to a specific type of manipulations, at the same time further fine-tuning can improve the robustness against new manipulations in the test set (pink line). The same conclusion holds in panel 10(b).

We also quantify the amount of manipulated data required for fine-tuning in order to improve the robustness of deep CAMA models. As shown in Figure 8, even using 1% of the manipulated data is sufficient to learn the vertical shift manipulation presented in the test set.

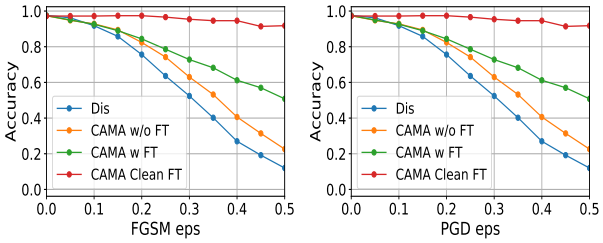
Adversarial Attack Test on MNIST We further test deep CAMA's robustness against two adversarial attacks: fast gradient sign method (FGSM) (Goodfellow et al., 2014) and projected gradient descent (PGD) (Madry et al., 2017). Note that, these attacks are specially developed for images with the small perturbation constraint. However, theses attack does not have guarantee to be valid by our definition as the manipulation depends on the class label Y, which has the risk of changing the ground-truth label. Such risk has also been discussed in Elsayed et al. (2018). Figure 11 show the results comparing deep CAMA and the DNN; both are trained on clean images only. Deep CAMA is significantly more robust to both attacks than the DNN (orange line), and with fine-tuning, deep CAMA shows additional $20\% - 40\%$ accuracy increase. We also show the clean data test accuracy after fine-tuning maintains to be the same thanks to our causal consistent model design.

Adversarial attack test on natural image classification We evaluate the adversarial robustness of deep CAMA when trained on natural images. In this case we follow Li et al. (2018) and consider *CIFAR-binary*, a binary classification dataset containing airplane and frog images from CIFAR-10. We choose to work with CIFAR-binary because VAE-based fully generative classifiers are less satisfactory for classifying clean CIFAR-10 images ($< 50\%$ clean test accuracy).



(a) FGSM (b) PGD

Figure 12. Adversarial robustness results on CIFAR-binary data.



(a) FGSM (b) PGD

Figure 13. Adversarial robustness results on measurement data.

The deep CAMA model trained with data augmentation (adding Gaussian noise with standard deviation 0.1, see objective (5)) achieves 88.85% clean test accuracy on CIFAR-binary, which is on par with the results reported in Li et al. (2018). For reference, a discriminative CNN with $2\times$ more channels achieves 95.60% clean test accuracy. Similar to previous sections we apply FGSM and PGD attacks with different ϵ values to both deep CAMA and the discriminative CNN, and evaluate classification accuracies on the adversarial examples before and after finetuning.

Results are reported in Figure 12. For both FGSM and PGD tests, we see that deep CAMA, before finetuning, is significantly more robust to adversarial attacks when compared with a discriminative CNN model. Regarding finetuning, although PGD with large distortion ($\epsilon = 0.2$) also fools the finetuning mechanism, in other cases finetuning still provides modest improvements (5% to 8% when compared with the vanilla deep CAMA model) without deteriorating test accuracy on clean data. Combined with adversarial robustness results on MNIST, we conjecture that with a better generative model on natural images the robustness of deep CAMA can be further improved.

4.2. Robustness test on measurement based data with generalized Deep CAMA

Our causal view on valid manipulations allows us to test model robustness on generic measurement data. Unfortunately, there exists no public multi-variate dataset where ground truth causal relationships are known. Therefore we generate synthetic data (see appendix) following a causal process, and test the performance of the generic deep CAMA on this measurement data. We use Gaussian variables for A, C and X, and categorical variables for Y. All the ground truth causal relationships are nonlinear (quadratic mainly).

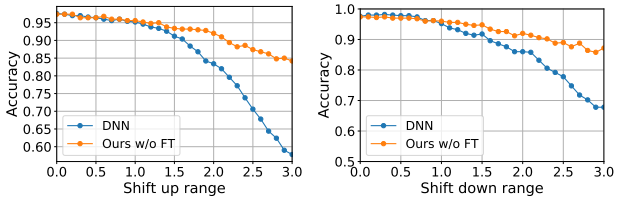
Manipulation Test First, we test manipulations on co-parents, C, while keeping the ground truth causal influence from C to X static. Thus, both C and X change. We manipulate C by shifting it up or down, which is a reasonable analogy to the noisiness in measurement data. For example, in medical measurement data, different doctors may have different subjective standards while examining the patients, thus the same measurement can be shifted up or down. Figure 14 shows the result: compared to a discriminatively trained DNN, deep CAMA is significantly more robust to a wide range of manipulations. However, when the range of the shifting manipulations increases, the classification accuracy of the discriminative DNN drops drastically. This confirms our theory in Section 3.2 that manipulations in C do not affect the decision making of deep CAMA, and therefore our model is more robust to manipulation on co-parents as compared to discriminative DNNs.

Figure 15 shows the performance of the generic deep CAMA with shifted X, and the model only sees clean data at training time. While deep CAMA achieves the same accuracy as a discriminative DNN on clean data, it is again significantly more robust to manipulations even without fine-tuning (the orange line vs the blue line). This robustness is further improved by fine-tuning (green line), especially when the amount of distortion is large. The red line shows that deep CAMA's test accuracy on clean data, which does not drop after fine-tuning on different shifts. This further confirms that during test time, fine-tuning learns the influence of M without affecting the causal relationships between Y and Z.

Adversarial Attack Test Lastly we evaluate the adversarial robustness of the generalized CAMA model. We only allow attacks on the children X and co-parents C according to our definition of valid attacks. This applies to both DNN and CAMA. Figure 13 shows the results in terms of test accuracy with adversarial examples generated using FGSM and PGD attack methods. Again deep CAMA demonstrate significantly improved robustness against adversarial attacks, and fine-tuning further provides improvements on robustness while keeping high accuracy on clean test examples.

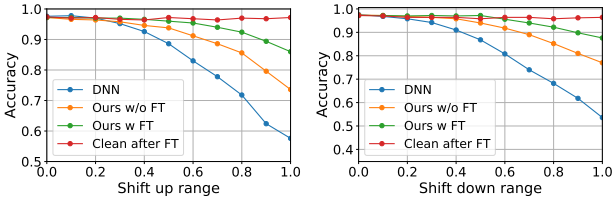
5. Related Work

Adversarial robustness Adversarial attacks can easily fool a discriminative DNN for vision/speech/language modelling tasks by adding imperceptible perturbations (Carlini



(a) Shift Up (b) Shift Down

Figure 14. Manipulate co-parents



(a) Shift Up (b) Shift Down

Figure 15. Manipulate children

& Wagner, 2018; Alzantot et al., 2018; Carlini & Wagner, 2017b; Szegedy et al., 2013; Papernot et al., 2017). Adversarial training (Madry et al., 2017; Tramèr et al., 2018) has shown some success in defending attacks, however, these techniques assume the knowledge of the adversary and present the perturbation to the model during training. Still, a discriminative model after adversarial training is vulnerable to unseen manipulations. Deep generative modelling has recently been applied as a defence mechanism to adversarial attacks. Specifically, existing work considered de-noising adversarial examples before feeding these inputs to the discriminative classifier (Song et al., 2018; Samangouei et al., 2018). Very recently, research revisited (deep) generative classifiers and provided evidence that they are more robust to adversarial attacks (Li et al., 2018; Schott et al., 2019; Lee et al., 2018).

Causal learning Causal inference has a long history in statistical research (Spirtes et al., 2000; Pearl, 2009; Peters et al., 2017; Pearl & Mackenzie, 2018). Although it has fundamental importance, the causal view has not been widely incorporated to the robustness analysis of neural networks on unseen manipulations. The most relevant work is in applying the existing causal views to transfer learning and domain adaption (Zhang et al., 2013; Stojanov et al., 2019; Zhao et al., 2019; Gong et al., 2016), where the difference in various domains are treated as either target shift or conditional shift from a causal perspective. As an extension to the domain adaptation work, Rothenhäusler et al. (2018); Heinze-Deml & Meinshausen (2017); Arjovsky et al. (2019) also discussed learning robust predictors across different domains. However, in these approaches the domain is specified either explicitly or though exemplar paired points, thus an unseen manipulation is not explicitly considered. In addition, interventions are considered only on a dataset level in domain adaption tasks, whereas we consider interventions on single data points, thus addressing a more general problem. By contrast, our proposed method does not rely on any given domain information. Another related area is causal feature selection (Aliferis et al., 2010), where causal discovery is applied first and features in the Markov Blanket of the prediction target are selected. We also note that CAMA's design is aligned with causal and anti-causal learning analyses (Schölkopf et al., 2012; Kilbertus et al., 2018), in that CAMA models the causal mechanism $Y \to X$ and use Bayes' rule for anti-causal prediction. Different from Schölkopf et al. (2012), CAMA is not limited to only two endogenous variables; rather it provides more generic design handling latent causes that correspond to both intrinsic variations and data manipulations.

Disentangled representations Learning disentangled representations has become a trendy research topic in recent representation learning literature. Considerable effort went to developing training objectives, e.g. β-VAE (Higgins et al., 2017) and other information theoretic approaches (Kim & Mnih, 2018; Chen et al., 2018). Additionally, different factorization structure in graphical model design has also been explored for disentanglement (Narayanaswamy et al., 2017; Li & Mandt, 2018). The deep CAMA model is motivated by the causal process of data manipulations, which differs from the model used in Narayanaswamy et al. (2017) in that the latent variables have different meanings. See appendix for a detailed discussion. Furthermore test-time fine-tuning allows deep CAMA to better adapt to unseen manipulations, which is shown to be useful for improving robustness.

6. Discussion

We have provided a causal view on the robustness of neural networks, showing that the vulnerability of discriminative DNNs can be explained by the lack of causal reasoning. We defined valid attacks under this causal view, which are intervention of the data though its causal factors which are not the target label or the ancestor of the target label. of the target variables, independent of the target and/or the cause of the target. We further proposed a deep causal manipulation augmented model (deep CAMA), which follows the causal relationship in the model design, and can be adapted to unseen manipulations at test time. Our model has demonstrated improved robustness, even without adversarial training. When manipulated data are available, our model's robustness increases for both seen and unseen manipulation.

Our framework is generic, however, manipulations can

change over time, and a robust model should adapt to these perturbations in a continuous manner. Our framework thus should be adapted to online learning or continual learning settings. In future work, we will explore the continual learning setting of deep CAMA where new manipulations come in a sequence. In addition, our method is designed for generic class-independent manipulations, therefore a natural extension would consider class-dependent manipulations where M is an effect of Y or there is a confounder for M and Y. Lastly out design excludes gradient-based adversarial attacks which is dependent on both the target and the victim model. As such attacks are commonly adopted in machine learning, we would also like to extend our model to such scenarios.

References

Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S., and Koutsoukos, X. D. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11(Jan):171–234, 2010.

Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017a.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017b.

Carlini, N. and Wagner, D. Audio adversarial examples: Targeted attacks on speech-to-text. *arXiv preprint arXiv:1801.01944*, 2018.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 2610–2620, 2018.

Elsayed, G. F., Shankar, S., Cheung, B., Papernot, N., Kurakin, A., Goodfellow, I., and Sohl-Dickstein, J. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*, 10, 2018.

Freeman, W. T. The generic viewpoint assumption in a framework for visual perception. *Nature*, 368(6471):542, 1994.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pp. 2839–2848, 2016.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Gopnik, A., Glymour, C., Sobel, D. M., Schulz, L. E., Kushnir, T., and Danks, D. A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3, 2004.

Heinze-Deml, C. and Meinshausen, N. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, volume 3, 2017.

Kilbertus, N., Parascandolo, G., and Schölkopf, B. Generalization in anti-causal learning. *arXiv preprint arXiv:1812.00524*, 2018.

Kim, H. and Mnih, A. Disentangling by factorising. In *International Conference on Machine Learning*, pp. 2654–2663, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

Li, Y. and Mandt, S. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pp. 5656–5665, 2018.

Li, Y., Bradshaw, J., and Sharma, Y. Are generative classifiers more robust to adversarial attacks? *arXiv preprint arXiv:1802.06552*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Narayanaswamy, S., Paige, T. B., Van de Meent, J.-W., Desmaison, A., Goodman, N., Kohli, P., Wood, F., and Torr, P. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pp. 5925–5935, 2017.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pp. 506–519. ACM, 2017.

Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Juang, Y.-L., Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., and Long, R. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. Learning independent causal mechanisms. *arXiv preprint arXiv:1712.00961*, 2017.

Pearl, J. *Causality*. Cambridge university press, 2009.

Pearl, J. and Mackenzie, D. *The book of why: the new science of cause and effect*. Basic Books, 2018.

Perez, L. and Wang, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. Anchor regression: heterogeneous data meets causality. *arXiv preprint arXiv:1801.06229*, 2018.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on MNIST. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1EHOsC9tX.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., and Richardson, T. *Causation, prediction, and search*. MIT press, 2000.

Stojanov, P., Gong, M., Carbonell, J., and Zhang, K. Data-driven approach to multiple-source domain adaptation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3487–3496, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

Woodward, J. *Making things happen: A theory of causal explanation*. Oxford university press, 2005.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Zhang, K., Schölkopf, B., Muandet, K., and Wang, Z. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pp. 819–827, 2013.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

A. Derivation Details

A.1. The intervention ELBO

When training with clean data $\mathcal{D} = \{(x_n, y_n)\}$, we set the manipulation variable M to a null value, e.g. $do(m = 0)$. In this case we would like to maximise the log-likelihood of the *intervened* model, i.e.

$$\max_{\theta} \mathbb{E}_{\mathcal{D}}[\log p_\theta(x, y|do(m = 0))].$$

This log-likelihood of the intervened model is defined by integrating out the unobserved latent variable Z in the intervened joint distribution, and from do-calculus we have

$$\begin{aligned} \log p_\theta(x, y|do(m = 0)) &= \log \int p_\theta(x, y, z|do(m = 0))dz \\ &= \log \int p_\theta(x|y, z, m = 0)p(y)p(z)dz. \end{aligned} \tag{12}$$

A variational lower-bound (or ELBO) of the log-likelihood uses a variational distribution $q(z|\cdot)$

$$\begin{aligned} \log p_\theta(x, y|do(m = 0)) &= \log \int p_\theta(x|y, z, m = 0)p(y)p(z)\frac{q(z|\cdot)}{q(z|\cdot)}dz \\ &\geq \mathbb{E}_{q(z|\cdot)}\left[\log \frac{p_\theta(x|y, z, m = 0)p(y)p(z)}{q(z|\cdot)}\right]. \end{aligned} \tag{13}$$

The lower-bound holds for arbitrary $q(z|\cdot)$ as long as it is absolutely continuous w.r.t. the posterior distribution $p_\theta(z|x, y, do(m = 0))$ of the intervened model. Now recall the design of the inference network/variational distribution in the main text:

$$q_\phi(z, m|x, y) = q_{\phi_1}(z|x, y, m)q_{\phi_2}(m|x),$$

where ϕ_1 and ϕ_2 are the inference network parameters of the corresponding variational distributions. Performing an intervention $do(m = 0)$ on this q distribution gives

$$q_\phi(z|x, y, do(m = 0)) = q_{\phi_1}(z|x, y, m = 0).$$

Defining $q(z|\cdot) = q_{\phi_1}(z|x, y, do(m = 0))$ and plugging-in it to eq. (13) return the *intervention* ELBO objective (3) presented in the main text.

A.2. The ELBO for unlabelled test data

The proposed fine-tuning method in the main text require optimising the marginal log-likelihood $\log p_\theta(x)$ for $x \sim \tilde{\mathcal{D}}$, which is clearly intractable. Instead of using a variational distribution for the unobserved class label Y, we consider the variational lower-bound of $\log p_\theta(x, y)$ for all possible $y = y_c$:

$$\begin{aligned} \log p_\theta(x, y) &= \log \int p_\theta(x, y, z, m)dzdm \\ &= \log \int p_\theta(x, y, z, m)\frac{q_\phi(z, m|x, y)}{q_\phi(z, m|x, y)}dzdm \\ &\geq \mathbb{E}_{q_\phi(z,m|x,y)}\left[\log \frac{p_\theta(x, y, z, m)}{q_\phi(z, m|x, y)}\right] := \text{ELBO}(x, y). \end{aligned} \tag{14}$$

Since both logarithm and exponent functions preserve monotonicity, and for all $y_c, c = 1, ..., C$ we have $\log p_\theta(x, y_c) \geq \text{ELBO}(x, y_c)$, we have

$$\log p_\theta(x, y_c) \geq \text{ELBO}(x, y_c), \forall c \;\Rightarrow\; p_\theta(x, y_c) \geq \exp[\text{ELBO}(x, y_c)], \forall c$$

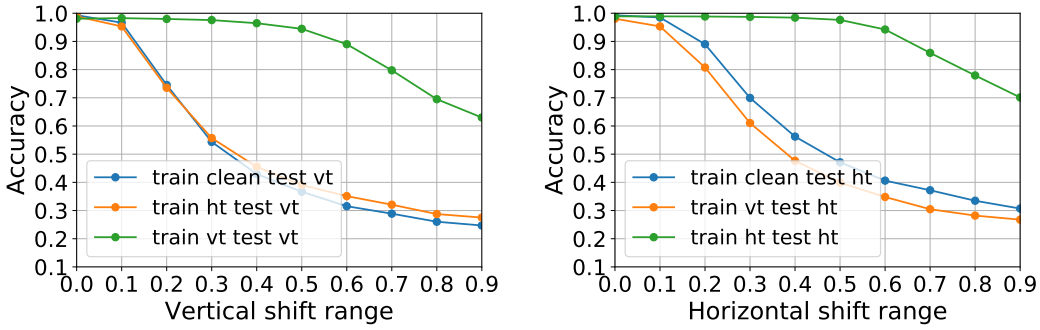
$$\Rightarrow\; \log p(x) = \log\left[\sum_{c=1}^{C} p_\theta(x, y_c)\right] \geq \log\left[\sum_{c=1}^{C} \exp[\text{ELBO}(x, y_c)]\right] := \text{ELBO}(x),$$

which justifies the ELBO objective (7) defined in the main text.

B. Additional Results

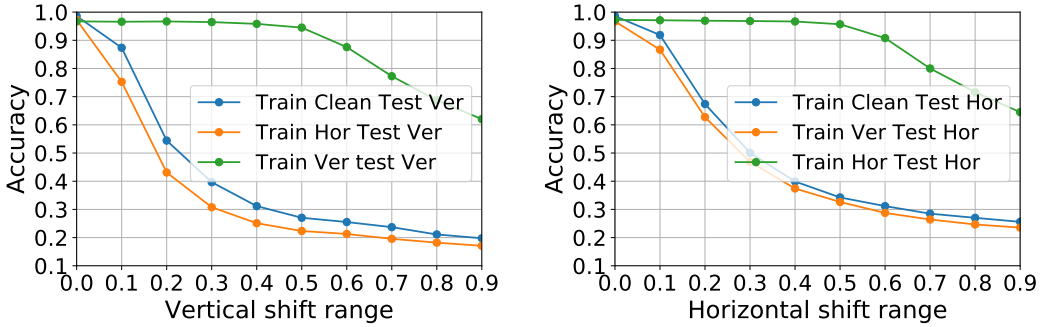
CNN We also performed experiments using different DNN network architectures. The convolution layers in CNN are designed to be robust to shifts. Thus, we test these vertical and horizontal shifts with a standard CNN architecture as used in https://keras.io/examples/cifar10_cnn/. 4 convolution layers are used in this architecture.

Figure 16 shows the performance against different shifts. We see that adding vertical shifts to the training data clearly harmed the robustness performances to unseen horizontal shifts as shown in 17(b). Adding horizontal shifted images in training did not influences the performance on vertical shifts much. Thus, we see that using different architectures of DNN, even the one that are designed to be robust to these manipulations, lack of generalization ability to unseen data is a common problem.



(a) Test Vertical shift (b) Test Horizontal shift

Figure 16. Robustness results for DNNs against different manipulations on MNIST using CNN. Panels (a) and (b) show the accuracy on classifying noisy test data generated by shifting the digits vertically (vt) and horizontally (ht). It shows that data augmentation during training makes generalization to unseen shifts worse (orange versus blue lines).



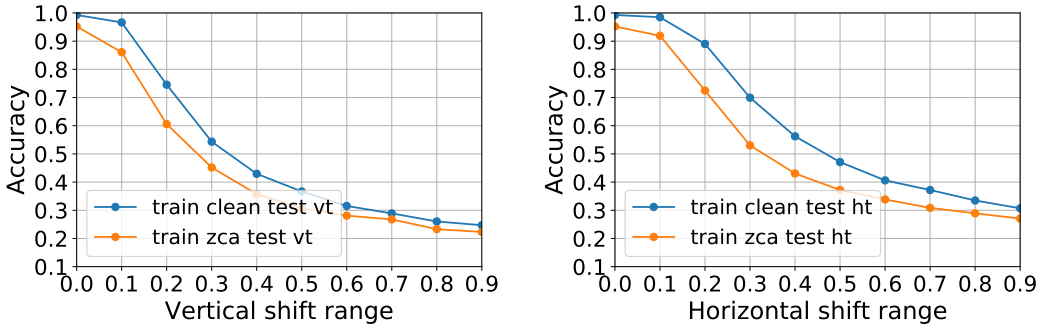
(a) Test Vertical shift (b) Test Horizontal shift

Figure 17. Robustness results for DNNs against different manipulations on MNIST using a large MLP. Panels (a) and (b) show the accuracy on classifying noisy test data generated by shifting the digits vertically (vt) and horizontally (ht). It shows that data augmentation during training makes generalization to unseen shifts worse (orange versus blue lines).

Enlarge Network Size Here we exam whether network capacity has any influence on the robustness performance to unseen manipulation. We use a wider network with [1024, 512, 512, 1024] units in each hidden layer instead of [512, 256, 126, 512] sized network in the paper. Figure 17 shows the robustness performance using this enlarged network. We observe the similar degree of over-fitting to the augmented data. The penalization ability shows no improvement by enlarging the network sizes.

ZCA Whitening Manipulation Our result does not limited to shifts, it generalizes to other manipulations. Figure 18 compare the result from training with clean images and training with ZCA whitening images added. We see that adding

ZCA whitening images in training harm both robustness against vertical shift and horizontal shift.



(a) Test Vertical shift (b) Test Horizontal shift

Figure 18. ZCA Whitening manipulation result. Figure shows the robustness results for DNNs against different manipulations on MNIST using CNN. The blue curve shows that result from training with clean data. The orange curve shows that result from training with zca whitening data added.

Additional Figures In addition to Figure 8, We also show the result testing with Vertical shift show in Figure 19, where a smaller N_M^p network ([dimM, 500, 500]) is used. The conclusion is the same was using the vertical shift. We need very few data for fine-tune. More than 1% data is sufficient.

Similar as Figure 8, we show the result using different percentage of data for fine-tuning in this experiment setting in 20.

B.1. Addition discussions on comparisons to Narayanaswamy et al. (2017)

Narayanaswamy et al. (2017) proposed a semi-supervised learning algorithm to learn disentangled representation for deep generative models. Their approach is generic, which only defines a joint distribution $p(x, y, z)$ as the model, and learn this model using the VAE approach. In their definition, X is the observed data, Y is a set of variables that are "interpretable" depending on the data context, and Z denotes the remaining implicit features. This means in their model Y is not limited to representing the prediction target; indeed their "intrinsic face" example the interpretable variables Y include lightning and shading of the face image. Then semi-supervised learning algorithm achieves disentanglement by assuming the existence of a few supervision signal for the "interpretable variables" Y.

On the other hand, in the proposed deep CAMA model the latent variables have very different meanings. Apart from the fact that Y is solely used to represent the prediction target, the M and Z variables are designed to separate the latent factors that

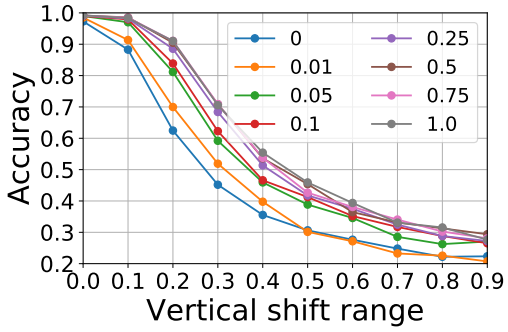


Figure 19. Performance regarding different percentage of test data used for fine-tuning manipulation of horizontal shift without using $do(m) = 0$ for the cleaning training data during fine-tuning.

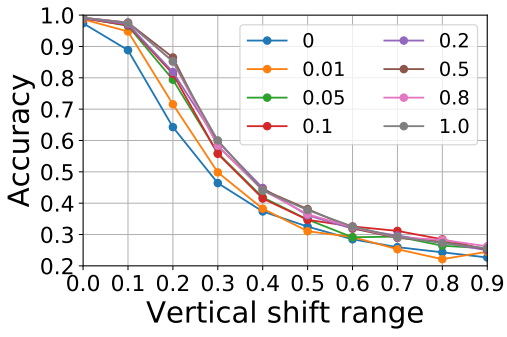


Figure 20. Performance regarding different percentage of test data used for fine-tuning manipulation of vertical shift using $do(m) = 0$ for the cleaning training data during fine-tuning.

can or cannot be *artificially* intervened by the adversary. Disentanglement of M and Z is achieved by training the model on interventional data (noisy data from valid manipulations). Furthermore the fine-tuning algorithm provides a test-time adaptation scheme for the deep CAMA model, which is different from the semi-supervised learning approach proposed by Narayanaswamy et al. (2017) which is conducted in training time. Importantly, the fine-tuning method updates the model parameter in a selective manner, which is motivated by our analysis on the causal generation process of noisy data.

C. Experimental settings

Network architecture

- MNIST experiments:
  - Discriminative DNN: The discriminate model used in the paper contains 4 densely connected hidden layer of $[512, 256, 126, 512]$ width for each layer. ReLU activations and dropout are used with dropout rate $[0.25, 0.25, 0.25, 0.5]$ for each layer.
  - Deep CAMA's p networks: we use $\text{dim}(Y) = 10, \text{dim}(Z) = 64$ and $\text{dim}(M) = 32$.
    NN^p_Y: an MLP of layer sizes $[\text{dim}(Y), 500, 500]$ and ReLU activations.
    NN^p_Z: an MLP of layer sizes $[\text{dim}(Z), 500, 500]$ and ReLU activations.
    NN^p_M: an MLP of layer sizes $[\text{dim}(M), 500, 500, 500, 500]$ and ReLU activations.
    $\text{NN}^p_{\text{merge}}$: an projection layer which projects the feature outputs from the previous networks to a 3D tensor of shape $(4, 4, 64)$, followed by 3 deconvolutional layers with stride 2, SAME padding, filter size $(3, 3, 64, 64)$ except for the last layer $(3, 3, 64, 1)$. All the layers use ReLU activations except for the last layer, which uses sigmoid activation.
  - Deep CAMA's q networks:
    NN^q_M: it starts from a convolutional neural network (CNN) with 3 blocks of $\{\text{conv}3 \times 3, \text{max-pool}\}$ layers with output channel size 64, stride 1 and SAME padding, then performs a reshape-to-vector operation and transforms this vector with an MLP of layer sizes $[4 \times 4 \times 64, 500, \text{dim}(M) \times 2]$ to generate the mean and log-variance of $q(m|x)$. All the layers use ReLU activation except for the last layer, which uses linear activation.
    NN^q_Z: first it uses a CNN with similar architecture as NN^M_q's CNN (except that the filter size is 5) to process x. Then after the reshape-to-vector operation, the vector first gets transformed by an MLP of size $[4\times4\times64, 500]$, then it gets combined with y and m and passed through another MLP of size $[500+\text{dim}(Y)+\text{dim}(M), 500, \text{dim}(Z)\times2]$ to obtain the mean and log-variance of $q(z|x, y, m)$. All the layers use ReLU activation except for the last layer, which uses linear activation.

- Measurement data experiments:
  - Discriminative DNN: The A, C, X variables are concatenated to an input vector of total dimension 20. Then the DNN contains 3 densely connected hidden layer of $[64, 16, 32]$ width for each layer, and output Y. ReLU activations and dropout are used with dropout rate $[0.25, 0.25, 0.5]$ for each layer.
  - Deep CAMA's p networks: we use $\text{dim}(Y) = 5, \text{dim}(A) = 5, \text{dim}(C) = 5, \text{dim}(Z) = 64$ and $\text{dim}(M) = 32$.
    $p(y|a)$: an MLP of layer sizes $[\text{dim}(A), 500, 500, \text{dim}(Y)]$, ReLU activations except for the last layer (softmax).
    $p(x|y, c, z, m)$ contains 5 networks: 4 networks $\{\text{NN}^p_Y, \text{NN}^p_C, \text{NN}^p_Z, \text{NN}^p_M\}$ to process each of the parents of X, followed by a merging network.
    NN^p_Y: an MLP of layer sizes $[\text{dim}(Y), 500, 500]$ and ReLU activations.
    NN^p_C: an MLP of layer sizes $[\text{dim}(C), 500, 500]$ and ReLU activations.
    NN^p_Z an MLP of layer sizes $[\text{dim}(Z), 500, 500]$ and ReLU activations.
    NN^p_M: an MLP of layer sizes $[\text{dim}(M), 500, 500, 500, 500]$ and ReLU activations.
    $\text{NN}^p_{\text{merge}}$: it first start from a concatenation of the feature outputs from the above 4 networks, then transforms the concatenated vector with an MLP of layer sizes $[500 \times 4, 500, \text{dim}(X)]$ to output the mean of x. All the layers use ReLU activations except for the last layer, which uses linear activation.
  - Deep CAMA's q networks:
    $q(m|x)$: it uses an MLP of layer sizes $[\text{dim}(X), 500, 500, \text{dim}(M) \times 2]$ to obtain the mean and log-variance. All the layers use ReLU activations except for the last layer, which uses linear activation.
    $q(z|x, y, m, a, c)$: it first concatenates x, y, m, a, c into a vecto, then uses an MLP of layer sizes $[\text{dim}(X) +$

$\dim(Y) + \dim(M) + \dim(A) + \dim(C), 500, 500, \dim(Z) \times 2]$ to transform this vector into the mean and log-variance of $q(z|x, y, m, a, c)$. All the layers use ReLU activations except for the last layer, which uses linear activation.

- CIFAR-binary experiments:
  - Discriminative CNN: The discriminate model used in the paper is a CNN with 3 convolutional layers of filter width 3 and channel sizes [128, 128, 128], followed by a flattening operation and a 2-hidden layer MLP of size $[4 \times 4 \times 128, 1000, 1000, 10]$. It uses ReLU activations and max pooling for the convolutional layers.
  - Deep CAMA's p networks: we use $\dim(Y) = 10, \dim(Z) = 128$ and $\dim(M) = 64$.
    NN^p_Y: an MLP of layer sizes $[\dim(Y), 1000, 1000]$ and ReLU activations.
    NN^p_Z: an MLP of layer sizes $[\dim(Z), 1000, 1000]$ and ReLU activations.
    NN^p_M: an MLP of layer sizes $[\dim(M), 1000, 1000, 1000]$ and ReLU activations.
    $\text{NN}^p_{\text{merge}}$: an projection layer which projects the feature outputs from the previous networks to a 3D tensor of shape $(4, 4, 64)$, followed by 4 deconvolutional layers with stride 2, SAME padding, filter size $(3, 3, 64, 64)$ except for the last layer $(3, 3, 64, 3)$. All the layers use ReLU activations except for the last layer, which uses sigmoid activation.
  - Deep CAMA's q networks:
    NN^q_M: it starts from a convolutional neural network (CNN) with 3 blocks of $\{\text{conv}3 \times 3, \text{max-pool}\}$ layers with output channel size 64, stride 1 and SAME padding, then performs a reshape-to-vector operation and transforms this vector with an MLP of layer sizes $[4 \times 4 \times 64, 1000, 1000, \dim(M) \times 2]$ to generate the mean and log-variance of $q(m|x)$. All the layers use ReLU activation except for the last layer, which uses linear activation.
    NN^q_Z: first it re-uses NN^q_M CNN network for feature extraction on x. Then after the reshape-to-vector operation, the vector gets combined with y and m and passed through another MLP of size $[4 \times 4 \times 64 + \dim(Y) + \dim(M), 1000, 1000, \dim(Z) \times 2]$ to obtain the mean and log-variance of $q(z|x, y, m)$. All the layers use ReLU activation except for the last layer, which uses linear activation.

Measurement data generation We set the target Y to be categorical, its children, co-parents and parents are continuous variables. The set 5 classes for Y, and Y has 10 children variables and 5 co-parents variables, also one 5 dimensional parents.

Parents (A) and co-parents (C) are generated by sampling from a normal distribution. We generate Y using structured equation $Y = f_y(A) + \sigma_Y$. We use $f_y = \text{argmax}\ g(A)$ and $g()$ is a quadratic function $0.2 * A^2 - 0.8A$. σ_Y is the Gaussain noise.

To generate the children $X = f(Y, C) + \sigma_x$, we also used quadratic function f and the parameters were sampled from a Gaussian distribution. As in the experiment, we were using fixed scale shift, we also added a normalize the children before adding the Gaussian random noise σ_x. So that all observations are in similar scale.

Other For MNIST experiments, we uses 5% of the training data as the validation set. We used the training results with the highest validation accuracy for testing. If not otherwise specified, 50% of noisy test data are used for fine-tuning in the shift experiments and all data are used for fine-tuning in the attack experiments.

For the experiments with measurement data. We generated 1000 data in total. We split, 500 data for testing, 450 for training and 50 for validation. We used the training results with the highest validation accuracy for testing for both deep CAMA and for DNN.