Classification with Rejection: Scaling Generative Classifiers with Supervised Deep Infomax

Xin Wang[*] and **Siu Ming Yiu**
The University of Hong Kong
{xwang, smyiu}@cs.hku.hk

Abstract

Deep Infomax (DIM) is an unsupervised representation learning framework by maximizing the mutual information between the inputs and the outputs of an encoder, while probabilistic constraints are imposed on the outputs. In this paper, we propose Supervised Deep InfoMax (SDIM), which introduces supervised probabilistic constraints to the encoder outputs. The supervised probabilistic constraints are equivalent to a generative classifier on high-level data representations, where class conditional log-likelihoods of samples can be evaluated. Unlike other works building generative classifiers with conditional generative models, SDIMs scale on complex datasets, and can achieve comparable performance with discriminative counterparts. With SDIM, we could perform classification with rejection. Instead of always reporting a class label, SDIM only makes predictions when test samples' largest class conditional surpass some pre-chosen thresholds, otherwise they will be deemed as out of the data distributions, and be rejected. Our experiments show that SDIM with rejection policy can effectively reject illegal inputs, including adversarial examples and out-of-distribution samples.

1 Introduction

Non-robustness of neural network models emerges as a pressing concern since they are observed to be vulnerable to adversarial examples [Szegedy *et al.*, 2013]. Many attack methods have been developed to find imperceptible perturbations to fool the target classifiers [Carlini and Wagner, 2017; Brendel *et al.*, 2017]. Meanwhile, many defense schemes have also been proposed to improve the robustnesses of the target models [Goodfellow *et al.*, 2014; Madry *et al.*, 2017].

An important fact about these works is that they focus on discriminative classifiers, which directly model the conditional probabilities of labels given samples. Another promising direction, which is almost neglected so far, is to explore robustness of generative classifiers [Ng and Jordan, 2002]. A generative classifier explicitly model conditional distributions of inputs given the class labels. During inference, it evaluates all the class conditional likelihoods of the test input, and outputs the class label corresponding to the maximum. Conditional generative models are powerful and natural choices to model the class conditional distributions, but they suffer from two big problems: (1) it is hard to scale generative classifiers on high-dimensional tasks, like natural images classification, with comparable performance to the discriminative counterparts. Though generative classifiers have shown promising results of adversarial robustness, they hardly achieve acceptable classification performance even on CIFAR10 [Li *et al.*, 2018; Schott and Rauber, 2018; Fetaya *et al.*, 2019]. (2) The behaviors of likelihood-based generative models can be counter-intuitive. They surprisingly assign higher likelihoods to out-of-distribution (OOD) samples [Nalisnick *et al.*, 2018; Choi and Jang, 2018]. [Fetaya *et al.*, 2019] discuss the issues of likelihood as a metric for density modeling, which may be the reason of non-robust classification, e.g. OOD samples detection.

In this paper, we propose supervised deep infomax (SDIM) by introducing *supervised statistical constraints* into deep infomax (DIM, [Hjelm *et al.*, 2018]), an unsupervised learning framework by maximizing the mutual information between representations and data. SDIM is trained by optimizing two objectives: (1) maximizing the mutual information (MI) between the inputs and the high-level data representations from encoder; (2) ensuring that the representations satisfy the supervised statistical constraints. The supervised statistical constraints can be interpreted as a generative classifier on high-level data representations giving up the *full* generative process. Unlike full generative models making implicit manifold assumptions, the supervised statistical constraints of SDIM serve as explicit enforcement of manifold assumption: data representations (low-dimensional) are trained to form clusters corresponding to their class labels. With SDIM, we could perform classification with rejection. SDIMs reject illegal inputs based on *off-manifold* conjecture [Samangouei *et al.*, 2018; Gu and Rigazio, 2014], where illegal inputs, e.g. adversarial examples, lie far away from the data manifold. Samples whose class conditionals are smaller than the pre-chosen thresholds will be deemed as *off-manifold*, and prediction requests on them will be rejected. The contributions of this paper are :

- We propose Supervised Deep Infomax (SDIM), an end-

[*]Contact Author

to-end framework whose probabilistic constraints are equivalent to a generative classifier. SDIMs can achieve comparable classification performance with similar discrinimative counterparts at the cost of small over-parameterization.

- We propose a simple but novel *rejection* policy based on *off-manifold* conjecture: SDIM outputs a class label only if the test sample's largest class conditional surpasses the pre-chosen class threshold, otherwise outputs *rejection*. The choice of thresholds relies only on training set, and takes no additional computations.
- Experiments show that SDIM with rejection policy can effectively reject illegal inputs, including OOD samples and adversarial examples generated by a comprehensive group of adversarial attacks.

2 Background: Deep InfoMax

Deep InfoMax (DIM, [Hjelm *et al.*, 2018]) is an unsupervised representation learning framework by maximizing the mutual information (MI) of the inputs and outputs of an encoder. The computation of MI takes only input-output pairs with the deep neural networks based estimator MINE [Belghazi *et al.*, 2018].

Let E_ϕ be an encoder parameterized by ϕ, working on the training set $\mathcal{X} = \{x_i\}_{i=1}^N$, and generating output set $\mathcal{Y} = \{E(x_i)\}_{i=1}^N$. DIM is trained to find the set of parameters ϕ such that: (1) the mutual information $\mathcal{I}(X, Y)$ is maximized over sample sets $\mathcal{X}$ and $\mathcal{Y}$. (2) the representations, depending on the potential downstream tasks, match some prior distribution. Denote $\mathbb{J}$ and $\mathbb{M}$ the joint and product of marginals of random variables X, Y respectively. MINE estimates a lower-bound of MI with Donsker-Varadhan [Donsker and Varadhan, 1983] representation of KL-divergence:

$$\begin{aligned}\mathcal{I}(X, Y) &= D_{KL}(\mathbb{J}||\mathbb{M}) \\ &\geq \mathbb{E}_{\mathbb{J}}[T_\omega(x, y)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_\omega(x,y)}]\end{aligned} \tag{1}$$

where $T_\omega(x, y) \in \mathbb{R}$ is a family of functions with parameters ω represented by a neural network. Since in representation learning we are more interested in *maximizing* MI, than its exact value, non-KL divergences are also favorable candidates. We can get a family of variational lower-bounds using f-divergence representations [Nguyen *et al.*, 2010]:

$$\mathcal{I}_f(X, Y) \geq \mathbb{E}_{\mathbb{J}}[T_\omega(x, y)] - \mathbb{E}_{\mathbb{M}}[f^*\big(T_\omega(x, y)\big)] \tag{2}$$

where f^* is the Fenchel conjugate of a specific divergence f. For KL-divergence, $f^*(t) = e^{(t-1)}$. A full f^* list is provided in Tab. 6 of [Nowozin *et al.*, 2016].

3 Supervised Deep InfoMax

All the components of SDIM framework are summurized in Fig. 1. The focus of Supervised Deep InfoMax (SDIM) is on introducing supervision to probabilistic constraints of DIM for (generative) classification. We choose to maximize the *local* MI, which has shown to be more effective in classification tasks than maximizing *global* MI [Hjelm *et al.*, 2018]. Equivalently, we minimize $\mathcal{J}_{\text{MI}}$:

$$\mathcal{J}_{\text{MI}} = -\frac{1}{M^2}\sum_{i=1}^{M^2} \tilde{\mathcal{I}}(L_\phi^{(i)}(\boldsymbol{x}), E_\phi(\boldsymbol{x})) \tag{3}$$

where $L_\phi(\boldsymbol{x})$ is a local $M \times M$ feature map of $\boldsymbol{x}$ extracted from some intermediate layer of encoder E, and $\tilde{\mathcal{I}}$ can be any possible MI lower-bounds.

3.1 Explicit Enforcement of Manifold Assumption

By adopting a generative approach $p(\boldsymbol{x}, y) = p(y)p(\boldsymbol{x}|y)$, we assume that the data follows the *manifold assumption*: the (high-dimensional) data lies on low-dimensional manifolds corresponding to their class labels. Denote $\tilde{\boldsymbol{x}}$ the compact representation generated with encoder $E_\phi(\boldsymbol{x})$. In order to explicitly enforce the manifold assumption , we admit the existence of data manifold in the representation space. Assume that y is a discrete random variable representing class labels, and $p(\tilde{\boldsymbol{x}}|y)$ is the real class conditional distribution of the data manifold given y. Let $p_\theta(\tilde{\boldsymbol{x}}|y)$ be the class conditionals we model parameterized with θ. We approximate $p(\tilde{\boldsymbol{x}}|y)$ by minimizing the KL-divergence between $p(\tilde{\boldsymbol{x}}|y)$ and our model $p_\theta(\tilde{\boldsymbol{x}}|y)$, which is given by:

$$\begin{aligned}D_{KL}\big(p(\tilde{\boldsymbol{x}}|y)||p_\theta(\tilde{\boldsymbol{x}}|y)\big) &= \mathbb{E}_{\tilde{\boldsymbol{x}},y\sim p(\tilde{\boldsymbol{x}},y)}[\log p(\tilde{\boldsymbol{x}}|y)] \\ &\quad - \mathbb{E}_{\tilde{\boldsymbol{x}},y\sim p(\tilde{\boldsymbol{x}},y)}[\log p_\theta(\tilde{\boldsymbol{x}}|y)]\end{aligned} \tag{4}$$

where the first item on RHS is a constant independent of the model parameters θ. Eq. 4 equals to maximize the expectation $\mathbb{E}_{\tilde{\boldsymbol{x}},y\sim p(\tilde{\boldsymbol{x}},y)}[\log p_\theta(\tilde{\boldsymbol{x}}|y)]$.

In practice, we minimize the following loss $\mathcal{J}_{\text{NLL}}$, equivalent to empirically maximize the above expectation over $\{\tilde{\boldsymbol{x}}_i = E_\phi(\boldsymbol{x}_i), y_i\}_{i=1}^N$:

$$\begin{aligned}\mathcal{J}_{\text{NLL}} &= -\mathbb{E}_{\tilde{\boldsymbol{x}},y\sim p(\tilde{\boldsymbol{x}},y)}[\log p_\theta(\tilde{\boldsymbol{x}}|y)] \\ &\approx -\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(\tilde{\boldsymbol{x}}_i|y_i)\end{aligned} \tag{5}$$

Besides the introduction of supervision, SDIM differs from DIM in its way of enforcing the statistical constraints: DIM use adversarial learningto push the representations to the desired priors, while SDIM directly maximizes the parameterized class conditional probability.

Maximize Likelihood Margins Since a generative classifier, at inference, decides which class a test input $\boldsymbol{x}$ belongs to according to its class conditional probability. On one hand, we maximize samples' true class conditional probabilities (classes they belong to) using $\mathcal{J}_{\text{NLL}}$; On the other hand, we also hope that samples' false class conditional probabilities (classes they do not belong to) can be minimized. This is assured by the following likelihood margin loss $\mathcal{J}_{\text{LM}}$:

$$\begin{aligned}\mathcal{J}_{\text{LM}} = \frac{1}{N} \cdot \frac{1}{C-1}\sum_{i=1}^{N}\sum_{c=1,c\neq y_i}^{C} &\max(\log p(\tilde{\boldsymbol{x}}_i|y = c) \\ &+ K - \log p(\tilde{\boldsymbol{x}}_i|y = y_i), 0)^2\end{aligned} \tag{6}$$

where K is a positive constant to control the margin. For each encoder output $\tilde{\boldsymbol{x}}_i$, the $C - 1$ true-false class conditional

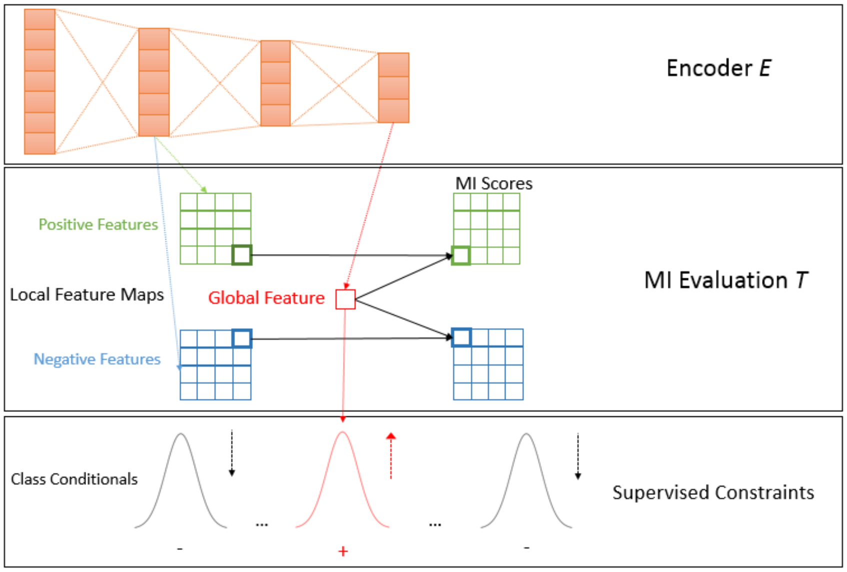

Figure 1: Components of SDIM framework. (1) The encoder E_ϕ takes input $\boldsymbol{x}$, and produces pairs of local feature maps $L_\phi(\boldsymbol{x})$ and global representations $E_\phi(\boldsymbol{x})$. (2) The MI evaluation network T_ω maps every possible positive pairs and negative pairs to MI scores specified by corresponding MI lower-bound. Negative pairs are simply obtained by combine all unpaired local feature maps and global representations within the same mini-batch. (3) Supervised constraints are imposed on the global representations $\tilde{\boldsymbol{x}} = E_\phi(\boldsymbol{x})$ for generative classification. The true class conditionals are maximized, while false class conditionals are minimized. See following parts of this section for details.

gaps are squared[1], which quadratically increases the penalties when the gap becomes large, then are averaged.

Putting all these together, the complete loss function we minimize is:

$$\mathcal{J}_{\text{SDIM}} = \alpha \cdot \mathcal{J}_{\text{MI}} + \beta \cdot \mathcal{J}_{\text{NLL}} + \gamma \cdot \mathcal{J}_{\text{LM}} \tag{7}$$

where α, β, γ are scaling factors.

Parameterization of Class Conditional Probability Each of the class conditional distribution is represented as an isotropic Gaussian. So the generative classifier is simply a embedding layer with C entries, and each entry contains the trainable mean and variance of a Gaussian. This *minimized* parameterization encourages the encoder to learn simple and stable low-dimensional representations that can be easily explained by even unimodal distributions. Considering that we maximize the true class conditional probability, and minimize the false class conditional probability at the same time, we do not choose conditional normalizing flows, since the parameters are shared across class labels, and the training can be very difficult. In [Schott and Rauber, 2018], each class conditional probability is represented with a VAE, thus scaling to complex datasets with huge number of classes, e.g. ImageNet, is almost impossible.

3.2 Decision Function with Rejection

A generative approach models the class-conditional distributions $p(\boldsymbol{x}|y)$, as well as the class priors $p(y)$. For classification, we compute the posterior probabilities $p(y|\boldsymbol{x})$ through Bayes' rule:

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})} \propto p(\boldsymbol{x}|y)p(y)$$

The prior $p(y)$ can be computed from the training set, or we simply use *uniform* class prior for all class labels by default. Then the prediction of test sample $\boldsymbol{x}^*$ from posteriors is:

$$y^* = \underset{c=[1\ldots C]}{\arg\max} \log p(\boldsymbol{x}^*|y = c). \tag{8}$$

The drawback of the above decision function is that it always gives a prediction even for illegal inputs. Instead of simply outputting the class label that maximizes class conditional probability of $\boldsymbol{x}^*$, we set a threshold for each class conditional probability, and define our decision function with rejection to be:

$$\begin{cases} y^*, & \text{if } \log p(\boldsymbol{x}^*|y^*) \geq \delta_{y^*} \\ \textit{Rejection}, & \text{otherwise} \end{cases} \tag{9}$$

The model gives a rejection when $\log p(\boldsymbol{x}^*|y^*)$ is smaller than the threshold δ_{y^*}. Note that here we can use $p(\boldsymbol{x}^*|y^*)$ and $p(\tilde{\boldsymbol{x}}^*|y^*)$ interchangeably.

Classification with rejection is not novel [Geifman and El-Yaniv, 2017], and previous works closely related to ours are the generative models based ones. The most recent one [Nalisnick *et al.*, 2019], which propose a hybrid model modeling distribution of features $p(\text{features})$ and predictive distribution $p(\text{targets}|\text{features})$ at the same time. Normalizing flow is used to learn invertible features as inputs of discriminative model, i.e. predictive distribution, and provides evaluation of features $\boldsymbol{x}^*$. Inputs out of the training data distribution are rejected by setting a threshold for $p(\boldsymbol{x}^*)$. For SDIM,

[1] Using squared margin, we achieve slightly better results in our experiments than simple margin.

illegal inputs are rejected by setting thresholds for each of the class conditional. The class conditionals are modeled on the data representations, rather than raw inputs. In terms of robustness, the hybrid model in [Nalisnick *et al.*, 2019] can successfully detect OOD samples, and applicability as well as performance on adversarial examples is not clear. While SDIM reject illegal inputs including OOD dataset samples and adversarial examples with more fine-grained class conditionals.

4 Experiments

Datasets We evaluate the effectiveness of the rejection policy of SDIM on four image datasets: MNIST, FashionMNIST (both resized to 32×32 from 28×28); and CIFAR10, SVHN. For out-of-distribution samples detection, we use the dataset pairs on which likelihood-based generative models fail [Nalisnick *et al.*, 2018; Choi and Jang, 2018]: FashionMNIST (in)-MNIST (out) and CIFAR10 (in)-SVHN (out). Adversarial examples detection are evaluated on MNIST and CIFAR10. Throughout our experiments, we use $\alpha = \beta = \gamma = 1$ in the loss function.

Choice of thresholds It is natural that choosing thresholds based on what the model knows, i.e. training set, and can reject what the model does not know, i.e. possible illegal inputs. We set one threshold for each class conditional. For each class conditional probability, we choose to evaluate on two different thresholds: *1st* and *2nd* percentiles of class conditional log-likelihoods of the correctly classified training samples. Compared to the detection methods proposed in [Li *et al.*, 2018], our choice of thresholds is much simpler, and takes no additional computations.

Models A typical SDIM instance consists of three networks: an encoder, parameterized by ϕ, which outputs a d-dimensional representation; mutual information evaluation networks, i.e. T_ω in Eqn. (1) and Eqn. (2); and C-way class conditional embedding layer, parameterized by θ, with each entry a $2d$-dimensional vector. We set $d = 64$ in all our experiments.

For encoder of SDIM, we use ResNet [He *et al.*, 2016] on 32×32 with a stack of $8n + 2$ layers, and 4 filter sizes $\{32$, 64, 128, $256\}$. The architecture is summarized as:

output map size	32×32	16×16	8×8	4×4
# layers	$1 + 2n$	$2n$	$2n$	$2n$
# filters	32	64	128	256

Table 1: Architecture summarization of the SDIM encoder.

The last layer of encoder is a d-way fully-connected layer. To construct a discriminative counterpart, we simply set the output size of the encoder's last layer to C for classification. We use ResNet10 ($n = 1$) on MNIST, FashionMNIST, and ResNet26 ($n = 3$) on CIFAR10, SVHN.

4.1 Evaluation on Clean Data

We report the classification accuracy (see Tab. 2 and Tab. 3) of SDIMs and the discriminative counterparts on clean test sets . Results show that SDIMs achieve the same level of accuracy as the discriminative counterparts with slightly increased number of parameters (17% increase for ResNet10, and 5% increase for ResNet26). We are aware of the existence of better results reported on these datasets , but pushing the state-of-the-art is not the focus of this paper.

Model	# Parameters	MNIST	Fashion
Disc. (ResNet10)	0.31M	99.42%	94.25%
SDIM (ResNet10)	0.36M (14% ↑)	99.55%	94.58%

Table 2: Performance of SDIMs and the discriminative counterparts on clean test sets of MNIST and FashionMNIST.

Model	# Parameters	CIFAR10	SVHN
Disc. (ResNet26)	4.39M	92.35%	95.96%
SDIM (ResNet26)	4.60M (5% ↑)	92.53%	95.74%

Table 3: Performance of SDIMs and the discriminative counterparts on clean test sets of CIFAR10 and SVHN.

Scalability

The most important fact about SDIM is it provides a end-to-end framework to train generative classifiers that achieve same-level performance as the corresponding discriminative counterparts. In contrast, despite the great success of fully conditional generative models in data synthesis, they perform poorly on classification tasks. In particular, they perform quite well on MNIST, but are still far away from achieving acceptable performance even on CIFAR10. For example, methods (GFZ & GFY) in *DeepBayes* [Li *et al.*, 2018] achieve $< 50\%$ accuracy, and they also mention that a conditional PixelCNN++ [Salimans *et al.*, 2017] (with much deeper networks) achieves 72.4% clean test accuracy. The test accuracy of ABS in [Schott and Rauber, 2018] is only 54%. Glow with class conditional mixture of Gaussian [Fetaya *et al.*, 2019] achieves 56.8%, could be improved to $80 - 85\%$ with reweighting or split prior. Many works have demonstrated that for classification tasks, discovering discriminative features (patches) is much more important than reconstructing the all the image pixels [Brendel and Bethge, 2019; Hjelm *et al.*, 2018]. The fact that methods in [Li *et al.*, 2018] improve the accuracy from $< 50\%$ to 92% by feeding the features learned by powerful discriminative classifier-VGG16 to their generative classifiers, also support this argument.

One direct effect of the poor classification performance of fully conditional generative models is that adversarial robustness evaluation on them is limited to MNIST, and becomes much less convincing on CIFAR10. A model with high test error implies that even for correctly classified samples, their distances to the decision boundaries are much small, thus are easy to craft small adversarial perturbations leading to misclassification [Gilmer *et al.*, 2018].

Furthermore, due to giving up the full generative process, SDIM is considerably smaller than fully conditional generative models. The GBZ of [Li *et al.*, 2018] on MNIST has

Dataset	Original Acc.	$1st$ Percentile		$2nd$ Percentile	
		Acc. Left	Rej. Rate	Acc. Left	Rej. Rate
MNIST	99.55%	99.95%	3.02%	99.97%	4.00%
FashionMNIST	94.58%	96.45%	4.63%	96.94%	6.60%
CIFAR10	92.53%	96.18%	8.90%	96.60%	10.86%
SVHN	95.74%	97.43%	3.99%	98.00%	6.36%

Table 4: Classification performances of SDIMs using the proposed decision function with rejection. We report the rejection rates of the test sets and the accuracy on the left test sets (after rejection) for each threshold.

1.5M parameters[2]. The ABS model in [Schott and Rauber, 2018] is 0.85M[3]. While SDIM on MNIST is only 0.36M. Note that their models don't scale on CIFAR10. So if we only care about the generative classification performance with no need to generate samples, it is unnecessary to model the full generative process, and giving up saves us a lot of computational resources.

Decision with Rejection

Given pre-chosen thresholds, there are chances that *legal* inputs are wrongly rejected. Thus we also investigate the implications of the proposed rejection decision function with different thresholds on clean test sets. The results in Tab. 4 show that choosing a higher percentile as threshold will reject more prediction requests. At the same time, the classification accuracy of SDIMs on the left test sets become increasingly better. This demonstrate that our rejection method help SDIMs reliably reject the low-confidence requests, and avoid wrong predictions.

4.2 Adversarial Examples Detection

We compare SDIM to GBZ [Li *et al.*, 2018], which consistently performs the best in *Deep Bayes*. Note that here we investigate the inherent robustness of generative classifier, so we do not incorporate the specially designed methods *outside* of the classifiers for adversarial examples detection.

FGSM and PGD We evaluate SDIM on L_∞ versions of them. We find that SDIMs are much more robust against FGSM and PGD than baseline (see Fig. 2), since the gradients numerically vanish as a side effect of the likelihood margin loss $\mathcal{J}_{\text{LM}}$ of SDIM. Recall that the class conditionals are optimized to keep a considerable margin. Before evaluating the cross entropy loss, softmax is applied on the class conditionals $\log p(\boldsymbol{x}|c)$ to generate a even sharper distribution. So for the samples that are correctly classified, their losses are numerically zeros, and the gradient on inputs $\nabla J_{\boldsymbol{x}}(\boldsymbol{x}, y)$ are also numerically zeros. This phenomena is similar to what some defenses using gradient obfuscation want to achieve. Defensive distillation [Carlini and Wagner, 2016] masks the gradients of cross-entropy by increasing the temperature of softmax. However, obfuscated gradients [Athalye *et al.*, 2018] give a false sense of security. For CW attacks, which do not use cross-entropy, and operate on logits directly, this could be ineffective.

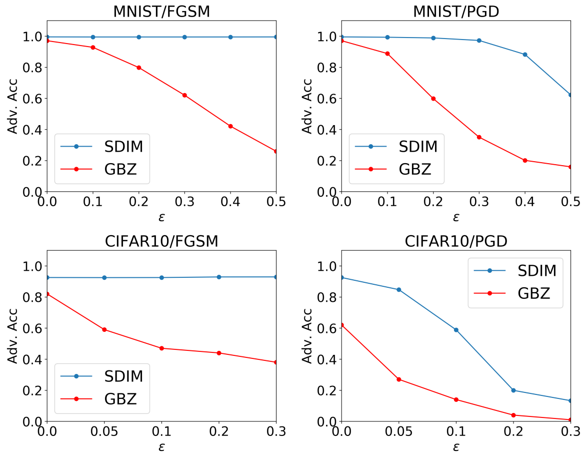

Figure 2: The adversarial classification accuracy of SDIM and GBZ on MNIST and CIFAR10 under FGSM-L_∞ and PGD-L_∞ attacks with different perturbation norm-bound ϵs. Note that GBZs don't scale on raw CIFAR10, results are models trained on features extracted VGG16.

CW Attack We evaluate SDIM and baselines on CW-$\mathcal{L}_2$ with loss factors $c = \{1, 10, 100, 1000\}$. On adversarial examples of MNIST, SDIM performs on par with the baseline, while on CIFAR10, the rejection/detection rate of SDIM is slightly better than the GBZ on simpler CIFAR10-binary (see Fig. 3). GBZ does not scale on CIFAR10, so the rejection rate is not available.

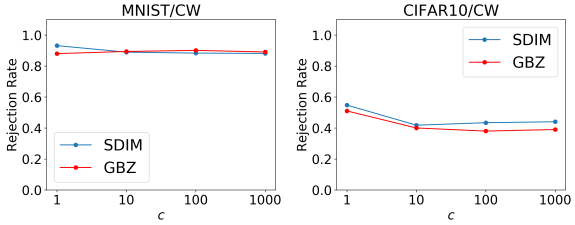

Figure 3: The rejection/detection rates of SDIM and GBZ on MNIST and CIFAR10. The rejection rate of GBZ (right) is on CIFAR10-binary (containing "airplane" and "frog" images from CIFAR-10) since it does not scale on full CIFAR10.

[2] The number of parameters is estimated based on the details in Appendix D of [Li *et al.*, 2018]. The encoder network (3 convs and 2-layer MLP) is $\sim$ 0.5M. $p(y|\boldsymbol{z})$ is 37k, and $p(\boldsymbol{x}|\boldsymbol{z})$ (2-layer MLP and 3 convs) is $\sim$1M.

[3] The number of parameters is calculated from the open sourced code: https://github.com/bethgelab/AnalysisBySynthesis.

Attacks	MNIST		CIFAR10	
	1st Per.	2nd Per.	1st Per.	2nd Per.
Boundary	100%	100%	100%	100%
LocalSearch	99.90%	100%	88.80%	93.10%

Table 5: Detection rates of SDIM models. We perform untargeted adversarial evaluation on the first 1000 images of test sets due to expensive computation.

Model	Fashion-MNIST	CIFAR10-SVHN
SDIM(*1st* Per.)	99.36%	94.24 %
SDIM(*2nd* Per.)	99.64%	95.81%
Glow(*10th* Per.)	3.53%	0.02%

Table 6: Mean detection rates of SDIMs and Glows with different thresholds on OOD detection.

Black-Box Attacks We also evaluate SDIM on black-box attacks: local search (score-based) attack [Narodytska and Kasiviswanathan, 2016], boundary(decision-based) attack [Brendel *et al.*, 2017]. Generally, we find that black-box adversarial examples are more easily to detect even on CIFAR10 (see Tab. 5).

Discussions on off-manifold conjecture [Gilmer *et al.*, 2018] challenges whether the off-manifold conjecture holds in general. They experiment on synthetic dataset-two high-dimensional concentric spheres with theoretical analyses, showing that even for a trained classifier with close to zero test error, there may be a constant fraction of the data manifold misclassified, which indicates the existence of adversarial examples *within* the manifold. But there are still several concerns to be addressed: First, as also pointed out by the authors, the manifolds in natural datasets can be quite complex than that of simple synthesized dataset. [Fetaya *et al.*, 2019] draws similar conclusion from analyses on synthesized data with particular geometry. So the big concern is whether the conclusions in [Gilmer *et al.*, 2018; Fetaya *et al.*, 2019] still hold for the manifolds in natural datasets. A practical obstacle to verify this conclusion is that works modeling the full generative processes are based on manifold assumption, but provide no explicit manifolds for analytical analyses like [Gilmer *et al.*, 2018; Fetaya *et al.*, 2019]. While SDIM enables explicit and customized manifolds on high-level data representations via probabilistic constraints, thus enables analytical analyses. In this paper, samples of different classes are trained to form isotropic Gaussians corresponding to their classes in representation space (other choices are also possible). The relation between the adversarial robustness and the forms and dimensionalities of data manifolds is to be explored in the future. Second, in their experiments, all models evaluated are discriminative classifiers. Considering the recent promising results of generative classifiers against adversarial examples, would using generative classifiers lead to different results? One thing making us feel optimistic is that even though the existence of adversarial examples is inevitable, [Gilmer *et al.*, 2018] suggest that adversarial robustness can be improved by minimizing the test errors, which is also supported by our experimental differences on MNIST and CIFAR10.

4.3 Out-Of-Distribution Samples Detection

Class-wise OOD detections are performed, and mean detection rates over all in-distribution classes are reported in Tab. 6. For each in-distribution class c, we evaluate the log-likelihoods of the whole OOD dataset. Samples whose log-likelihoods are lower the class threshold δ_c will be detected as OOD samples. Same evaluations are applied on conditional Glows with *10th* percentile thresholds, but the results are not good. The results are clear and confirm that SDIMs, generative classifiers on high-level representations, are more effective on classification tasks than fully conditional generative models on raw pixels. Note that fully generative models including VAE used in [Li *et al.*, 2018; Schott and Rauber, 2018] fail on OOD detection [Nalisnick *et al.*, 2018; Choi and Jang, 2018]. The stark difference of SDIM from full generative models (flows or VAEs) is that SDIM models samples' likelihood in the high-level representation spaces, rather than directly on the raw pixels.

Summary SDIM models perform on par with or better than strongest variant GBZ in [Li *et al.*, 2018] on detection of various types of adversarial examples. However, the performance of SDIM could extend to complex datasets, while GBZ is limited to MNIST. Modeling likelihood on image representations, SDIM models easily detect OOD samples. While *fully* generative models, e.g. GBZ, who model likelihood on raw image pixels, are known to fail on this task.

5 Conclusions and Future Directions

Though some promising results of the robustness of generative classifiers have been observed, it is challenging to scale them on complex datasets. In this paper, we introduce supervised probabilistic constraints to DIM. Giving up the full generative process, SDIMs are equivalent to generative classifiers on high-level data representations. Unlike full conditional generative models which achieve poor classification performance even on CIFAR10, SDIMs attain same-level performance as the comparable discriminative counterparts on complex datasets. The training of SDIM is also computationally similar to discriminative classifiers, and does not require prohibitive computational resources. Our proposed rejection policy based on *off-manifold* conjecture, a built-in defense mechanism of SDIM, can effectively reject illegal inputs, including adversarial examples, and OOD samples. We demonstrate that likelihoods modeled on high-level data representations, rather than raw pixel intensities, are more robust on downstream tasks without the requirement of generating real samples.

The rejection mechanism in this paper is different but complementary to other defense mechanisms, e.g. adversarial training. Performing *classification with rejection*, classifiers can refuse to make low-confidence predictions, while adversarial training aims to inherently improve models' recognition robustness. It is to be explored how to combine them to build more trustworthy models.

References

[Athalye *et al.*, 2018] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[Belghazi *et al.*, 2018] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[Brendel and Bethge, 2019] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019.

[Brendel *et al.*, 2017] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[Carlini and Wagner, 2016] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[Carlini and Wagner, 2017] 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.

[Choi and Jang, 2018] Hyunsun Choi and Eric Jang. Generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[Donsker and Varadhan, 1983] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

[Fetaya *et al.*, 2019] Ethan Fetaya, Jacobsen, and Richard Zemel. Conditional generative models are not robust. *arXiv preprint arXiv:1906.01171*, 2019.

[Geifman and El-Yaniv, 2017] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pages 4878–4887, 2017.

[Gilmer *et al.*, 2018] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Gu and Rigazio, 2014] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hjelm *et al.*, 2018] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[Li *et al.*, 2018] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? *arXiv preprint arXiv:1802.06552*, 2018.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[Nalisnick *et al.*, 2018] Eric Nalisnick, Akihiro Matsukawa, and et al. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[Nalisnick *et al.*, 2019] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. *arXiv preprint arXiv:1902.02767*, 2019.

[Narodytska and Kasiviswanathan, 2016] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[Ng and Jordan, 2002] Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

[Nguyen *et al.*, 2010] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[Nowozin *et al.*, 2016] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

[Salimans *et al.*, 2017] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[Samangouei *et al.*, 2018] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[Schott and Rauber, 2018] Lukas Schott and Jonas et al Rauber. Towards the first adversarially robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.