PERCEPTUAL REGULARIZATION: VISUALIZING AND LEARNING GENERALIZABLE REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

A deployable machine learning model relies on a good representation. Two desirable criteria of a good representation are to be understandable, and to generalize to new tasks. We propose a technique termed perceptual regularization that enables both visualization of the latent representation and control over the generality of the learned representation. In particular our method provides a direct visualization of the effect that adversarial attacks have on the internal representation of a deep network. By visualizing the learned representation, we are also able to understand the attention of a model, obtaining visual evidence that supervised networks learn task-specific representations. We show models trained with perceptual regularization learn transferrable features, achieving significantly higher accuracy in unseen tasks compared to standard supervised learning and multi-task methods.

1 INTRODUCTION

Identifying useful features is the cornerstone of a good computer vision model (Rumelhart et al., 1986; Bengio et al., 2013). Classical feature extraction strategies such as the SIFT features (Lowe, 2004) and SURF features (Bay et al., 2006) are handcrafted and rely on human expertise. The success of deep learning is in replacing human-engineered features by automatically learned features (Donahue et al., 2014; Zhou et al., 2015; 2016). However, this breakthrough has brought a new set of problems.

First, the features learned by deep neural networks are very vulnerable, small changes known as adversarial attacks can completely change the model's prediction. There is a rapidly growing body of work on adversarial examples in the context of deep networks (Szegedy et al., 2014; Nguyen et al., 2015; Bubeck et al., 2019; Ford et al., 2019). They raise many security concerns, such as the reliability of driverless cars, or the trustworthiness of facial recognition systems (Papernot et al., 2016). Robustness to such adversarial attacks is becoming a common requirement (Madry et al., 2017). Further understanding of what adversarial attacks do to a model's representation will help move these objectives forward.

Second, the complex hierarchical nature of deep networks makes it inherently difficult to understand the flow of information between layers, and in particular what information a given layer's representation contains. This is of crucial importance for interpretation of the predictions of a model, which has had much effort and attention in last few years (Yosinski et al., 2015; Ribeiro et al., 2016; Doshi-Velez & Kim, 2017; Al-Shedivat et al., 2017; Melis & Jaakkola, 2018). Machine interpretability is, however, still an ongoing project.

Third, the features learned by a supervised deep network are known to be tailored - perhaps overly so - to the specific task they were trained on (Yosinski et al., 2014). This has led to incredible performances on supervised learning tasks, but has made the transferability of features to new tasks difficult, sometimes even harming performance (Pan & Yang, 2009; Rosenstein et al., 2005). It is still an open question to understand when features trained on one task will transfer to another.

In this paper, we introduce a framework, termed *perceptual regularization*, that gives insight into all of these questions. This is achieved by stacking a decoder on top of the model's latent representation with the aim of providing a human understandable reconstruction, see Figure 1. One of the key merits of perception regularization is its simplicity and naturalness. A simple regularization term is all that is needed to yield significant visual insight into a range of questions about the representations learned by deep networks.

The main contributions of this paper are:

- Introducing *perceptual regularization*, a method for jointly learning a visualization for deep representations, and for learning representations that are predictive on unseen tasks.
- Using the visualization method to shed light on what effect adversarial attacks have on a model.
- Using this method to visualize how the attention of a model focuses only on aspects that are relevant for prediction. More generally we give promising evidence to suggest that our visualization method is of interest for model diagnostics and interpretability.
- Showing that representations learned using perceptual regularization transfer effectively to tasks that were not known a priori.

2 PERCEPTUAL REGULARIZATION

We begin by introducing perceptual regularization. Consider a deep image classification model such as AlexNet (Krizhevsky et al., 2012), VGG-Net (Simonyan & Zisserman, 2015), or ResNet (He et al., 2016). It usually consists of two stages: several convolutional layers, followed by dense layers. The convolution layers can be viewed as a feature extraction process which we denote by $\mathcal{F}$, transforming images from the input space $\mathcal{X}$ into a latent space $\mathcal{Z}$. The dense layers $\mathcal{C} : \mathcal{Z} \to \mathcal{Y}$ act as a classifier on the latent space. In order to regularize and visualize the latent representation, we introduce a decoder $\mathcal{D} : \mathcal{Z} \to \mathcal{X}$ on top of the latent space $\mathcal{Z}$ which we jointly train with the classification model. The architecture is shown in Figure 1.

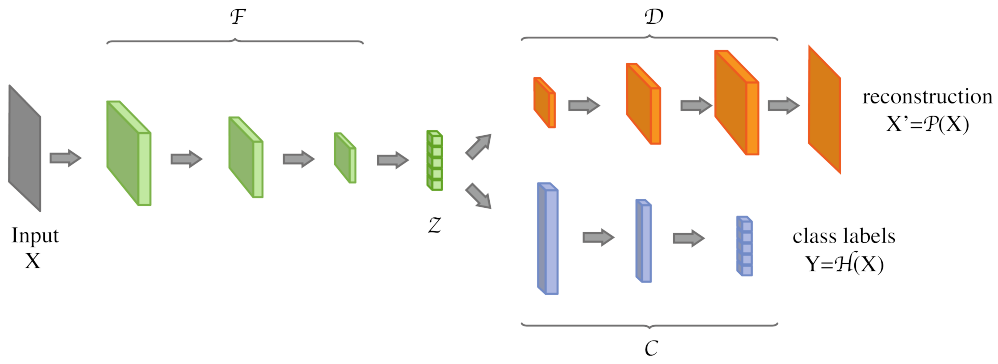


Figure 1: Perceptual regularization: we stack a decoder on top of the feature map and jointly train it with the classifier.

More precisely, we denote the classification model by $\mathcal{H} = \mathcal{C} \circ \mathcal{F}$ and term the reconstruction $\mathcal{P} = \mathcal{D} \circ \mathcal{F}$ a *perception* of the model. The regularization strategy we introduce is to jointly train $\mathcal{H}$ and $\mathcal{P}$ together by minimizing

$$\min_{\mathcal{P},\mathcal{H}} \mathbb{E}_{X,Y} \left[\underbrace{\ell(\mathcal{H}(X), Y)}_{\text{classification loss}} + \lambda \underbrace{\|X - \mathcal{P}(X)\|^2}_{\text{reconstruction loss}} \right]. \qquad \text{(Perceptual regularization)}$$

The first term is the standard classification loss and we use the cross-entropy loss throughout the paper. The regularization term is the ℓ_2 reconstruction loss scaled by a hyper-parameter $\lambda > 0$. Intuitively, the decoder tries to reconstruct the original image based on the same information the classifier uses for prediction. In particular, if the feature map $\mathcal{F}$ preserves all the information from the input, the perception should be able to perfectly recover X. Conversely, if a lot of information has been thrown away, then we will get a poor reconstruction and the regularization loss will be high. Therefore, our regularization biases the learned representation to maintain more information about the input compared to normal supervised learning. This will be useful if one would like to reuse the learned features later for other tasks.

Notice that the regularization term vanishes when $\mathcal{P}$ is the identity mapping, in which case the feature map $\mathcal{F}$ is bijective. This can happen when the model is very powerful and therefore able to memorize any dataset. However, this situation is less interesting in the sense that a good representation should efficiently compress the information. This idea has been clearly understood with the information

bottleneck objective, for which it is known that $\mathcal{F}(X)$ approximates the minimal sufficient statistic for Y given X (Shamir et al., 2010). Therefore, to constrain the expressiveness of the feature map, we impose a bottleneck structure in the network architecture. More precisely, the dimension of the latent space $\mathcal{Z}$ is made significantly smaller than the dimension of the input space $\mathcal{X}$. This enforces the feature map to compress information and makes it very unlikely to be bijective.

As well as regularizing the model, our perceptual regularization also provides a natural way to visualize the latent space. In particular, the perception $\mathcal{P}$ is designed to decode the feature map $\mathcal{F}$, which is very useful for model understanding. In the next two sections we begin to explore the insights that can be drawn using perceptual regularization.

3 How do adversarial attacks affect a classifier's perception?

Adversarial examples can seem quite mysterious, and point to a way in which human and machine perceptions are misaligned (Han et al., 2019; Engstrom et al., 2019). While intuitively reasonable, understanding of this phenomenon has been limited by the difficulty of observing the effect of adversarial attacks on the internal representation of a model (Zhang & Zhu, 2018; Olah et al., 2018). In this section we explore the first application of our perceptual regularization: to provide human understandable visualizations of a model's internal representation of adversarial examples.

First, we briefly recap how adversarial examples are obtained. Given a classification model $\mathcal{H}$ and a sample (X, Y), we aim to fool the model's prediction by slightly perturbing the input X. We constrain the perturbation to be in a small neighborhood $\Delta(X)$ of X, which is usually defined as an ℓ_p ball. Specifically, we are looking for

$$X_{\text{adv}} \in \underset{X' \in \Delta(X)}{\arg\max}\ \ell(\mathcal{H}(X'), Y) \tag{1}$$

While many different strategies have been proposed to adversarially attack the model (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Kurakin et al., 2017; Athalye et al., 2018) , we focus on the projected gradient descent (PGD) attack (Madry et al., 2017) throughout this paper. Specifically, we set $\Delta(X)$ to be an ℓ_∞ ball and the adversarial example X_{adv} is obtained by performing 100 iterations of PGD attack. Examples of adversarial attacks can be found in the first row of Figures 2.

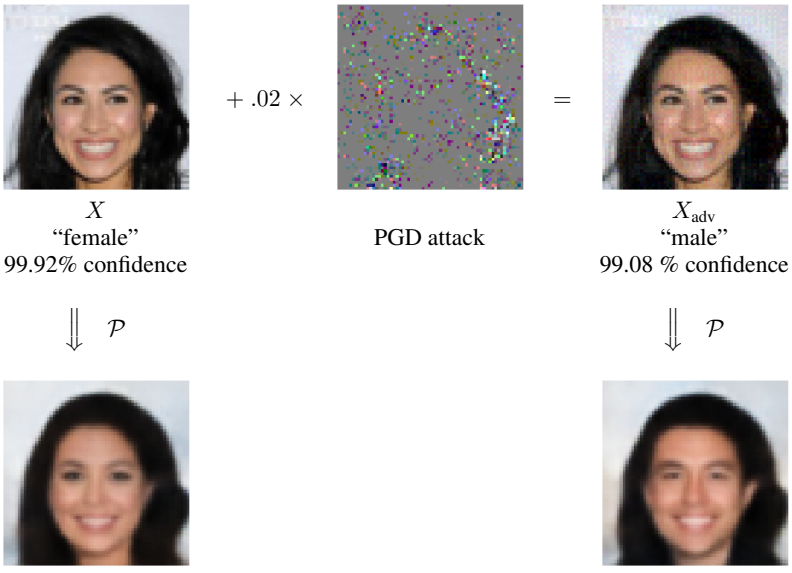


Figure 2: First row: by adding an imperceptibly small vector whose elements are obtained from performing a PGD attack, we can change the classification of the image. Second row: by applying perception, we can visualize why the model make the "incorrect" prediction.

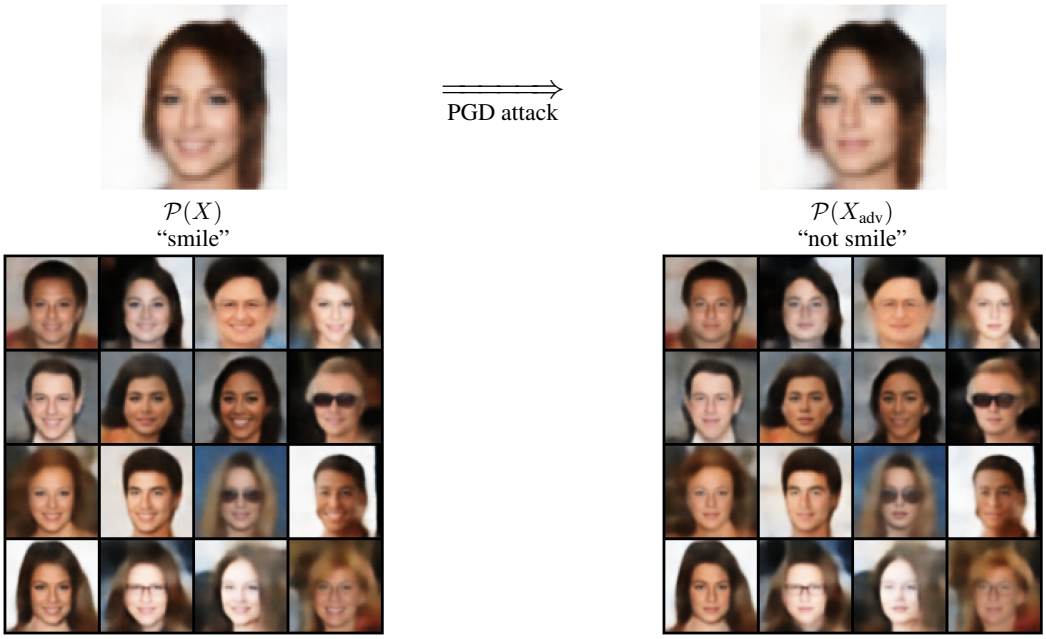


Figure 3: Images after applying perception on the task of classifying smiling face, we can visualize why the model make the "incorrect" prediction. The corresponding original images can be found in Appendix A.

As is well known, adversarial examples are hard to distinguish from the original image with human perception. Now, if we jointly train the classifier and decoder with perceptual regularization, besides the classification model, we also have in hand a perception branch that aims to transfer the latent variable to images. We adversarially attack the classification branch $\mathcal{H} = \mathcal{C} \circ \mathcal{F}$ using (1) to obtain X_{adv}. If, instead of viewing X and X_{adv}, we look at the model's perception of each image $\mathcal{P}(X)$ and $\mathcal{P}(X_{\text{adv}})$, a clear disparity emerges between how the model understands the two examples. What is extremely striking is that the model's perception and incorrect classification are semantically consistent to the human eye. To put it simply: *when the model misclassifies an image of a female as male, the perception of the adversarial image really looks like a male to a human*, as shown in Figure 2; *when the model misclassifies an image of a smiling face as not smiling, the perception of the adversarial image really looks like it is not smiling*, as shown in Figure 3.

We provide further experiments of our method on the MNIST (LeCun et al., 1998) and CelebA (Liu et al., 2015) datasets. Our perception decoder allows one to visualize the effect of an adversarial attack on a deep network's representation. It successfully reveals a clear difference in machine perception even though humans cannot distinguish the two. For CelebA we train three different classifiers for three different binary classification tasks (Eyeglasses/Smiling/Gender). All experiments support the same observations made in this section. See the appendix for many more examples.

This shows that given the model's feature representation, it is making a prediction that appears reasonable on adversarial images. In other words, the representation learned by the model is very different from human perception. It is also interesting that the images obtained from perception (Figure 3, in the second row of Figure 2,) are simultaneously close to each other and yet highly distinguishable in the characteristics most important for prediction. This gives interesting visual insight into the way in which adversarial attacks find a short path to a decision boundary. For example, Figure 2 has several subtle modifications: lighter eyes, squarer forehead and jaw, thicker nose. The changes are very targeted, capturing the essence of the global feature.

This begs the question: can adversarial attacks be used to identify and modify high-level features? Global features are high level human-interpretable features such as, "wearing glasses", "is smiling", or "is female"). There has been much previous work on learning disentangled representations for which the latent variables correspond to disentangled human-interpretable features (Tenenbaum & Freeman, 2000; Chen et al., 2016; Higgins et al., 2017). To then modify a given global feature one needs to figure out which latent feature (or which combination) to adjust. However, the task of finding the "correct" latent direction is non-trivial.

Adversarial attacks as global feature adjustment. Our visualization method yields a novel way for user-controlled global feature adjustment. The recipe is as follows:

- Train a model with perceptual regularization,
- Given X, obtain X_{adv} by performing an adversarial attack toward the target label (e.g. smiling/wear eyeglasses),
- Apply perception $\mathcal{P}$ on the adversarial example to obtain the adjusted image $\mathcal{P}(X_{\text{adv}})$.

Figures 2 and 3 give examples of this. We believe it is possible to obtain more realistic images using recent progress on GANs, but we leave this for future work. Finally, we stress that our method requires labeled data while GANs do not. However, if one has labeled data, our method provides a promising alternative since it does not require human intervention.

4 VISUALIZING FEATURES LEARNED BY A DEEP NETWORK

A second application of perceptual regularization is to visualize what a classifier's attention focuses on. In particular, if a model is trained to predict the object in the middle of an image, it is commonly believed that the learned representation "overfits" to the training task by focusing only on the central part of the image. This can be a cause of negative transfer - when reusing a learned representation for a new task actually *hurts* performance (Pan & Yang, 2009; Kuzborskij & Orabona, 2013). We show our visualization method can help understand when a representation will cause negative transfer.

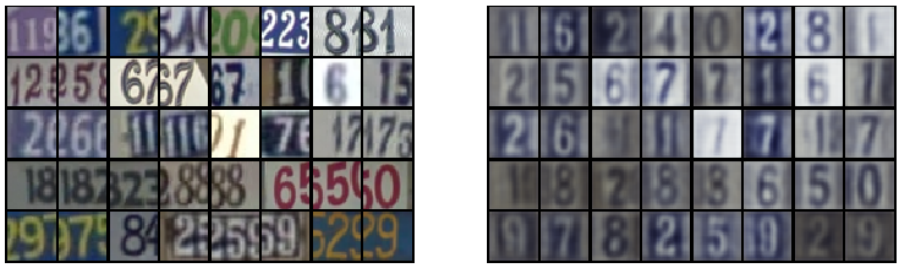


Figure 4: Left: Original images from SVHN. Right: The model's perception of the images.

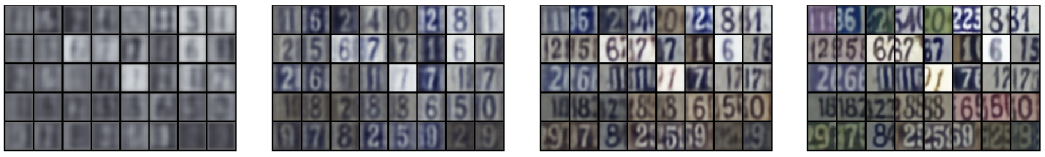
4.1 THE ATTENTION PROBLEM

It has long been intuitively understood that the representations in the final layers of a supervised deep network will only focus on aspects of the data relevant to its prediction task (Long et al., 2015; Yosinski et al., 2014). In this section we demonstrate on the SVHN dataset (Netzer et al., 2011) that this phenomenon of focused attention can be visualized using perceptual regularization.

The SVHN dataset consists of images of house numbers and the task is to predict the number in the center of the image. In particular, an image could contain multiple numbers but the label is given according to the middle one. A natural question is: will a supervised model learn a representation that remembers numbers that are not in the center? To answer this question, we train a CNN with perceptual regularization and then use perception to visualize the attention of the model, i.e. given any input X, we visualize $\mathcal{P}(X)$. As shown in Figure 4, our visualization technique shows that the model learns a representation that focuses on the middle of the image, and forgets what is at the edges of the image. This provides very strong visual evidence to support the common belief that a supervised model learns a representation specific to its prediction task. This is fine if there is only one task of interest, but makes transferring features between tasks problematic. For example, in our case, the representation would be completely ineffective for identifying the digit on the left or right hand sides.

4.2 PERCEPTUAL REGULARIZATION TO CONTROL ATTENTION

The cause of this limited attention problem is the supervised learning paradigm itself. In this section, we demonstrate that perceptual regularization can be useful to reduce the problem of learning overly specific features. In particular, our method allows for a smooth interpolation between supervised learning and unsupervised learning. In the extreme case, when $\lambda \to 0$, we recover the standard supervised learning setting, and when $\lambda \to \infty$ we recover the classical auto-encoder objective. Hence, the parameter λ can control how much information we would like our model's representation to keep.



(a) $\lambda = 10^{-4}$, Acc: 88.5%. (b) $\lambda = 10^{-3}$, Acc: 87.9%. (c) $\lambda = 10^{-2}$, Acc: 87.8%. (d) $\lambda = 10^{2}$, Acc: 81.3%.

Figure 5: Visualizing the model's attention. Acc is the accuracy of the classifier on the test set.

We run experiments on the SVHN dataset by varying the regularization parameter λ. We keep the architecture fixed (see Appendix B.3 for more details). Figure 5 shows different regimes of features for different values of λ. There are four main regimes:

- Tiny ($\lambda < 10^{-4}$, Figure 5(a)): Since λ is extremely small the contribution to the loss from the decoder branch is so negligible that there is not much incentive to learn a good decoder and the visualization breaks down.
- Delicate ($10^{-4} < \lambda < 10^{-2}$, Figure 5(b)): A good decoder is learned and the bias injected into the learned feature map by the regularization term is small. The number in the center is well reconstructed, but the edges of the image are not and the colors are missing. It is in this regime that we are able to see the restricted attention of a classifier.
- Intermediate ($10^{-2} < \lambda < 1$, Figure 5(c)): We obtain more faithful reconstructions compared to the "delicate" regime. In particular, the numbers at the edges are reasonably reconstructed but colors are still not perfect.
- Large ($\lambda > 1$, Figure 5(d)): The decoder is very good, both numbers and colors are recovered. The regularization has had a significant effect on the nature of the learned features. Since these features retain much more information than for smaller λ they are more likely to transfer well to new unseen tasks.

Clearly the choice of λ heavily affects the features learned by the model. The larger λ is, the more information is preserved in the features: small values of λ only preserve the number in the middle of the image, intermediate values can also reconstruct the boundary of the image, and only for large λ is the input's precise color correctly reconstructed. It is also important to note that the accuracy on the supervised task decreases as λ increases, but at a relatively slow rate. This is to be expected. In the limit $\lambda \to \infty$ there is no longer any incentive to perform good classification, only reconstruction. An interesting future direction will be to understand why certain features are harder to learn than others, for instance, in our example why the color is recovered later than numbers along the edges.

It is important to mention that the representation learned using perceptual regularization is not exactly the same as what is learned without it. This is because the reconstruction loss encourages representations to preserve as much information as possible about the original input. If one wanted to avoid this bias then the obvious solution is to learn the feature map first, then learn the visualization decoder separately, with the feature map frozen. This strategy works reasonably well when the network's size is small. But, as the network gets more complicated it becomes computationally difficult to decode the feature map. Hence, as an alternative solution, we train the decoder jointly with the feature map. What we have shown is that although the bias induced by the regularization is an apparent drawback, it can in fact be turned into a strength by using λ to control the behavior of the feature map.

Finally, the specific values of λ quoted are meant only to serve as a very rough guide about trends. Making the magnitude of the reconstruction loss and classification loss comparable is often effective to locate the "delicate" λ.

5 Towards obtaining generalizable features for unseen tasks

We call a representation generalizable if it is useful for tasks beyond the one(s) it was trained on. In the previous section, we gave visual evidence that features learned from supervised learning are task related and so may not be generalizable. In contrast, our proposed perceptual regularization encourages the feature to capture more details of the original input, which may help generalizability. In this section we build on these observations by showing that features learned using perceptual regularization are indeed more predictive on new, unseen tasks.

Our approach is as follows: given access to training data for a source task S, we train a model for task S and obtain a feature map $\mathcal{F}$ and a classifier $\mathcal{C}_S$. Then, when a new target task T comes along, we use the same features $\mathcal{F}$ learned previously, but train a new classifier $\mathcal{C}_T$ on top of it. In other words, we freeze the feature representation and fine-tune the last layers.

For experiments, we used the CelebA dataset. We considered three different image understanding tasks: identifying if a face has Glasses, is Smiling, and the individual's Gender. We compare our methods to standard supervised training (with ℓ_2 regularization). The architecture is kept exactly the same and the only difference is the regularization. The results can be found in Figures 6 and 7. The results are averaged over 6 trials. For full experimental details see the appendix.

S Source \ T Target	Eyeglasses	Smile	Gender
Eyeglasses	**95.63 %**	59.32 %	74.30%
Smile	58.49%	**89.12 %**	63.43%
Gender	71.34%	60.38%	**93.86%**

S Source \ T Target	Eyeglasses	Smile	Gender
Eyeglasses	**87.55%**	83.03%	86.47%
Smile	85.96%	**84.17%**	86.26%
Gender	86.13%	82.63%	**87.09%**

Figure 6: Left: Standard supervised training with weight decay. Right: Our method: perceptual regularization with $\lambda = 10^{-3}$.

Supervised learning (Figure 6 Left): As one would expect, supervised learning methods have strong performance when $S = T$ (the diagonal). But these features do not transfer well to new tasks, with accuracy falling significantly when $S \neq T$ (the off-diagonal). This is to be expected since standard supervised training learns features tailored specifically to the training task, as illustrated in Section 4.

Our method (Figure 6 Right): In contrast, the results for perceptual regularization (with $\lambda = 10^{-3}$) show that the learned features generalize to new tasks extremely well, with comparable accuracies across all tasks. Indeed, for an unseen target task T, the accuracy varies by at most two percentage points. When compared to standard supervised learning features, perceptual regularization achieves between $12 - 30\%$ better accuracy when $S \neq T$. The cost is that our method performs slightly poorer than the standard training method when $S = T$. This is simply due to balancing task specific attention and the generality we are aiming for.

For further comparison, we benchmark our results against multi-task learning methods with hard parameter sharing (Caruana, 1997), see Figure 7. In the multi-task setting, the model have access to multiple datasets during training. The model jointly trains the same feature map for the training tasks, but uses distinct classifiers so as to maximize the *average* empirical risk across the tasks. The results reported are the accuracies for fine tuning the learned representation for a task T. The families of training tasks are as follows: Multi2-1 is Smile and Gender; Multi2-2 is Eyeglasses, and Gender; Multi2-3 is Eyeglasses and Smile. Multi3 uses all three datasets for training. In other words, there is no unseen dataset, and T is part of the training dataset.

Multi-task learning (Figure 7): We emphasize that this setting is strictly stronger than our setting because the model has the opportunity to see multiple datasets during training (two or three in our experiments) which should help learn features that can transfer between tasks. Despite this unfair comparison, our method still outperforms this approach on unseen tasks. When the multi-task training involved data from the test task T the multi-task results are better (for the same reasons as when $S = T$ in the supervised setting).

S Source \ T Target	Eyeglasses	Smile	Gender
Multi2-1	62.15%	94.60%	93.34%
Multi2-2	89.06%	73.39%	93.67%
Multi2-3	89.41%	95.24%	74.53%
Multi3	88.76%	93.94 %	93.15%

Figure 7: Multi-task supervised training.

6 DISCUSSION AND CONCLUDING REMARKS

In this paper we introduced *perceptual regularization*, a method for visualizing and learning representations that can generalize to unseen tasks. To achieve this, our method combines a generative and discriminative model. There has been previous work on the idea of combining generative and discriminative models (Lasserre et al., 2006; Grabner et al., 2007; Larochelle & Bengio, 2008; Le et al., 2018), however these works have focused on the idea of using such a model to exploit semi-supervised (unlabeled) data. To the best of our knowledge our work is the first to identify the generative plus discriminative formulation as an effective method for practical, visual appraisal of the representation learned by a model. We use this visualization to understand adversarial examples and study the attention of a model. We also identify this formulation as suitable for learning representations that are effective on previously unseen tasks.

In this work we made the choice to consider the ℓ_2 loss as a metric between images. However, the specific nature of the injected bias is particular to ℓ_2: it permits lots of small errors, but penalizes large differences severely. It would be an interesting future direction to consider different metrics and consider what kind of representations are learned by regularizing with different generative model objectives such as GANs, VAEs, or InfoMAX (Goodfellow et al., 2014; Kingma & Welling, 2014; Hjelm et al., 2019).

Another interesting question is to understand what conditions are necessary to obtain good visualizations. We identify two possible factors. First: model complexity. As the task becomes more challenging we may need a more powerful decoder and refined training strategy (e.g. applying methods from the GAN literature). Second: human meaningful features. There has been work suggesting that more complicated datasets have small but highly predictive features that humans do not notice (Tsipras et al., 2018). The model may learn to use these features for prediction. In this case our visualizations may also not be human meaningful. Nevertheless, our visualization method could be a useful certificate for whether the model learns human meaningful features.

This raises the question of how to learn robust features that generalize across multiple tasks. There is evidence suggesting adversarial training may help learn more "human-aligned" features. But so far work on adversarial training has mainly focused on supervised learning. We suspect that this will not be sufficient to learn transferrable features. It would be an interesting next step to combine perception regularization with adversarial training.

Our visualizations raise a broader point about the supervised learning paradigm: models insist on pigeon-holing every input into one of a fixed number of classes, no matter how out of sample the input is. Consider the following toy example: taking X to be random noise and computing the adversarial attack X_{adv} for a model $\mathcal{H}$ trained on MNIST, then $\mathcal{P}(X_{\text{adv}})$ looks like a digit to the human eye (see Figure 8). This is undesirable and it is not sufficient to simply add an 11th "other" class. This is a long way from the dynamic way that humans create and learn new classes of objects.

To conclude, we have shown that perceptual regularization is a promising approach to answering many important questions in machine learning. However we believe this is just the beginning, and view perceptual regularization as a broadly useful tool for model diagnostics with many more uses to be uncovered in the future.

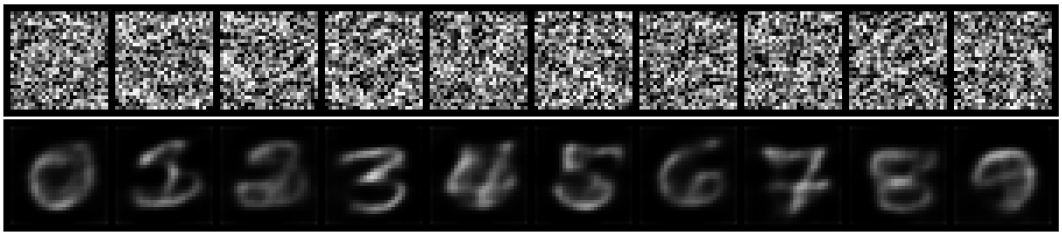


Figure 8: First row: X_{adv} for random noise X. Second row: $\mathcal{P}(X_{\text{adv}})$. Even for what is essentially random noise, the model insists on perceiving something that semantically looks like a number.

REFERENCES

M. Al-Shedivat, A. Dubey, and E. P. Xing. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conferences on Machine Learning (ICML)*, 2018.

H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pp. 404–417. Springer, 2006.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

S. Bubeck, Y. T. Lee, E. Price, and I. Razenshteyn. Adversarial examples from computational constraints. In *International Conferences on Machine Learning (ICML)*, 2019.

R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

J. Donahue, Y. Jia, J. Vinyals, O.and Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655, 2014.

F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Learning perceptually-aligned representations via adversarial robustness. *arXiv preprint arXiv:1906.00945*, 2019.

N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

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

H. Grabner, P. Roth, and H. Bischof. Eigenboosting: Combining discriminative and generative information. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2007.

C. Han, W. Yoon, G. Kwon, S. Nam, and D. Kim. Representation of white-and black-box adversarial examples in deep neural networks and humans: A functional magnetic resonance imaging study. *arXiv preprint arXiv:1905.02422*, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

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

R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

I. Kuzborskij and F. Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pp. 942–950, 2013.

H. Larochelle and Y. Bengio. Classification using discriminative restricted boltzmann machines. In *International Conferences on Machine Learning (ICML)*, 2008.

J. Lasserre, C. Bishop, and T. Minka. Principled hybrids of generative and discriminative models. *IEEE conference on computer vision and pattern recognition (CVPR)*, 2006.

L. Le, A. Patterson, and M. White. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. December 2015.

M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *International Conferences on Machine Learning (ICML)*, 2015.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

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

D. A. Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2015.

C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 2016.

M. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pp. 3, 2005.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Nature*, 323:533–536, 1986.

O. Shamir, S. Sabato, and N. Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural computation*, 12(6):1247–1283, 2000.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

Q. Zhang and S. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. In *International Conference on Learning Representations (ICLR)*, 2015.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

A ADDITIONAL IMAGES ON ADVERSARIAL EXAMPLES

Figure 9: Visualization of adversarial examples on CelebA dataset with Smile labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

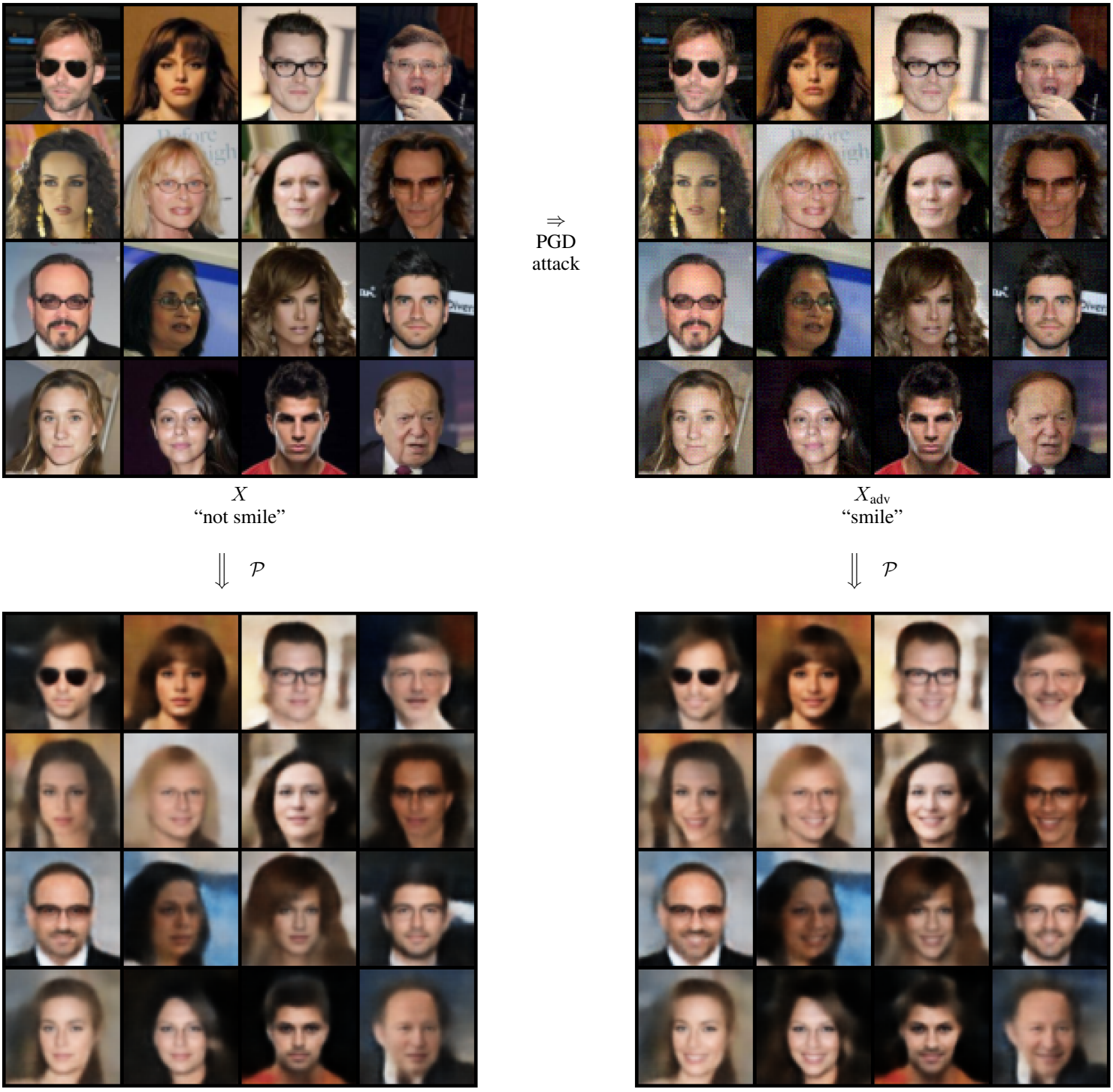


Figure 10: Visualization of adversarial examples on CelebA dataset with Smile labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

Figure 11: Visualization of adversarial examples on CelebA dataset with eyeglasses labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

Figure 12: Visualization of adversarial examples on CelebA dataset with Gender labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

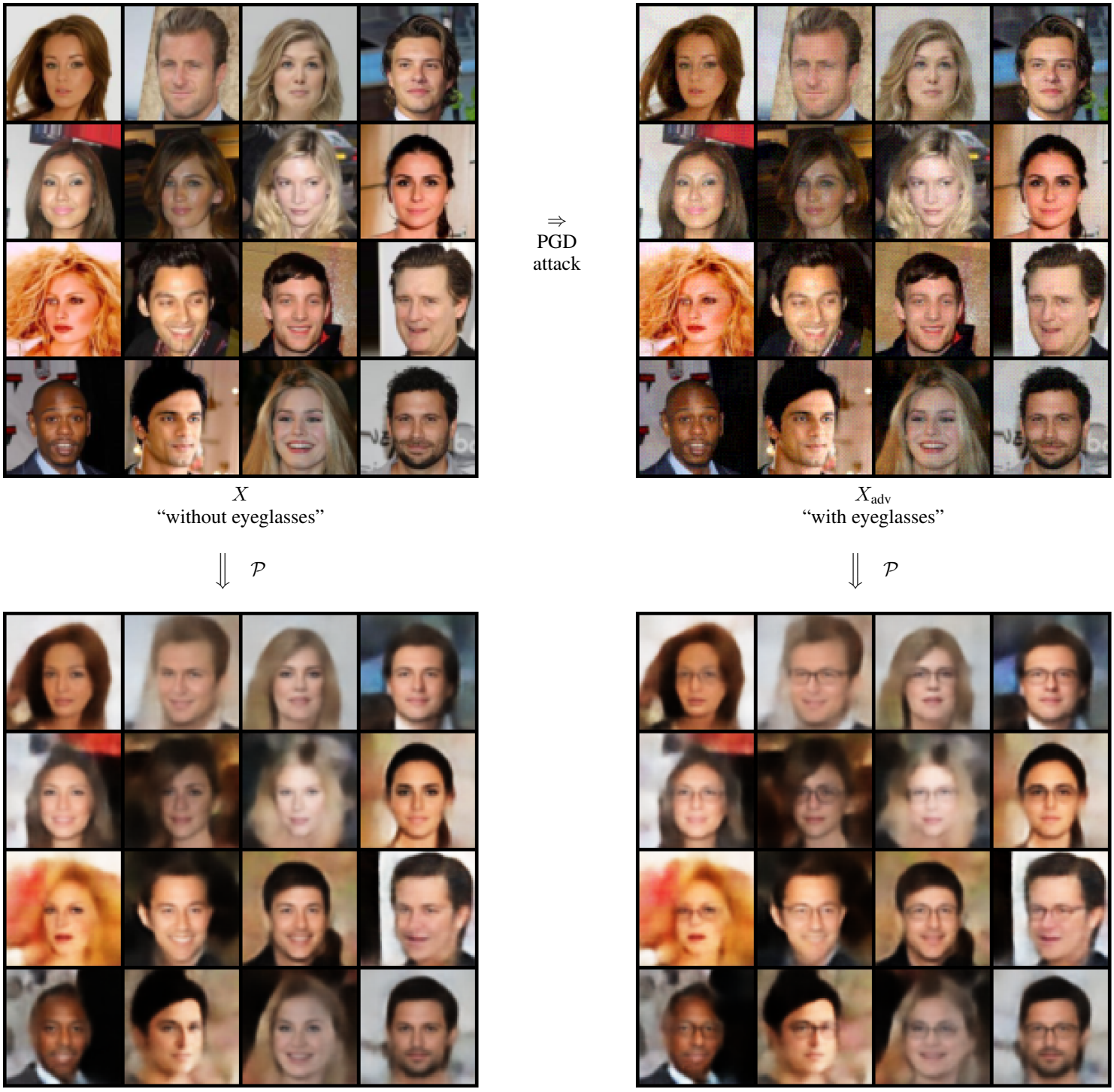


Figure 13: Visualization of adversarial examples on CelebA dataset with Eyeglasses labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

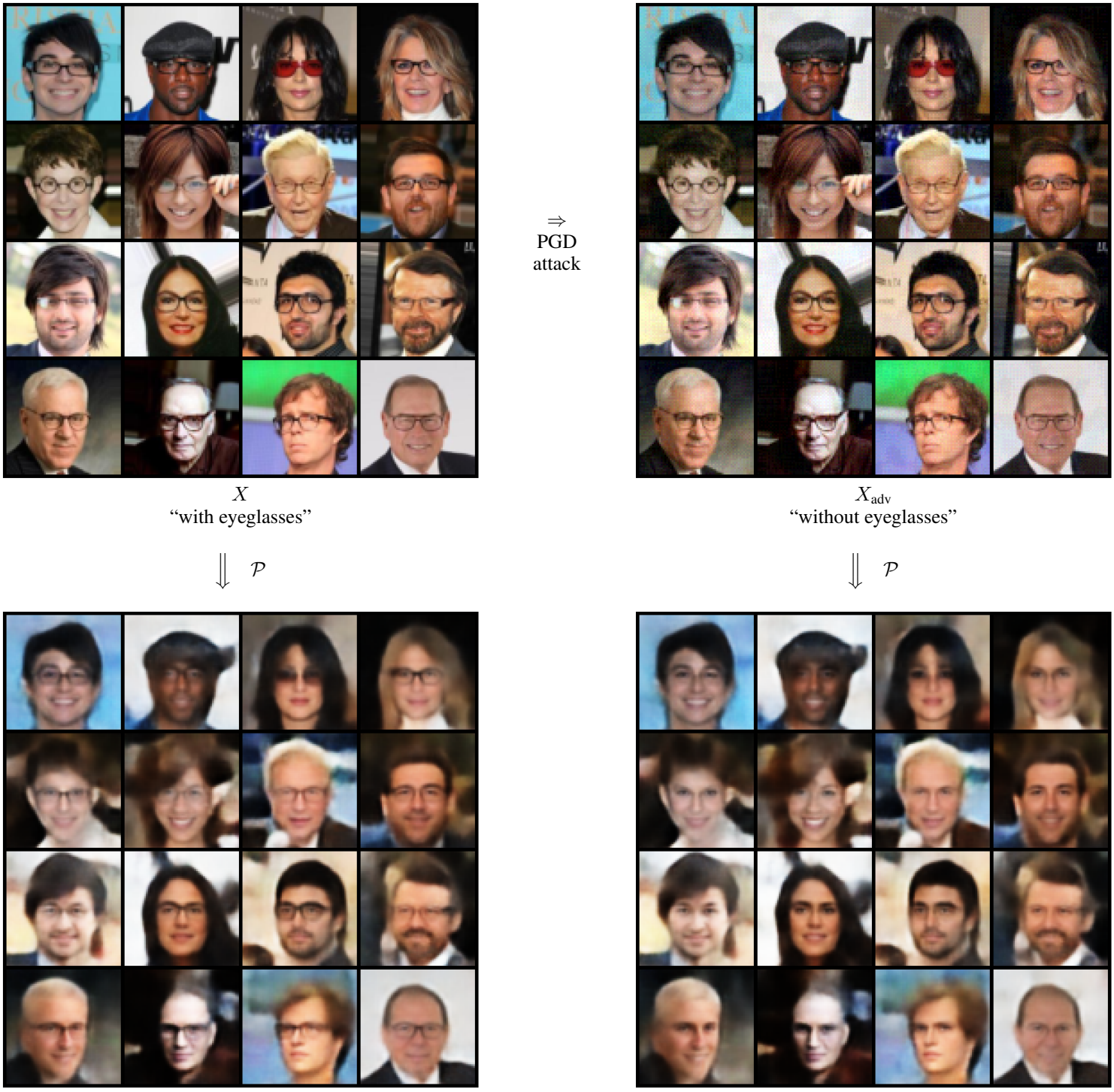


Figure 14: Visualization of adversarial examples on CelebA dataset with Eyeglasses labels. The adversarial examples are obtained by applying 100 iterations of PGD attack with ℓ_∞ perturbation and $\epsilon = 0.03$. The regularization parameter is set to $\lambda = 10^{-3}$.

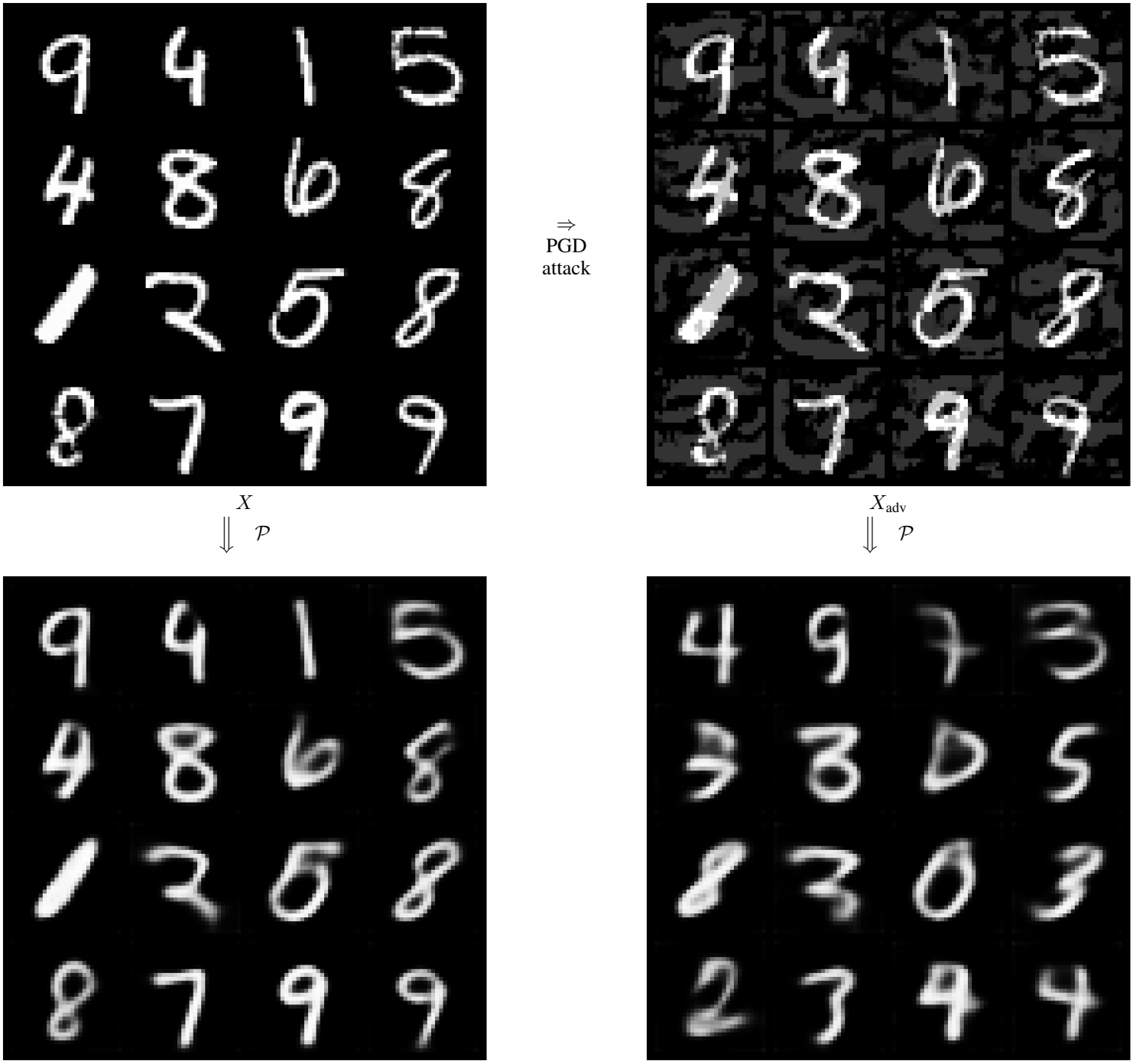


Figure 15: Visualization of adversarial examples on MNIST dataset with $\epsilon = 0.2$ (ℓ_∞ perturbation). The regularization parameter is set to $\lambda = 10^{-2}$.

B ARCHITECTURE DETAILS

B.1 EXPERIMENTAL SETTING ON CELEBA

For the CelebA experiments, we resize the images to 64×64 and we subsample the dataset to obtain a relatively balanced dataset on the labels we are considering : Eyeglasses/Smile/Gender. The statistics of the training set and testing set are as follows:

	Size	Eyeglasses	Smile	Gender
Training Set	20000	50%	44.72%	59.60%
Test Set	6386	50%	43.83%	59.51%

Figure 16: The percentage indicates the proportion of positive labels among the dataset.

We use Adam optimizer (Kingma & Ba, 2015) with $\beta = (0.9, 0.999)$, learning rate 10^{-3} in the first 20 iterations and 10^{-4} for later iterations. No data augmentation is used. The architecture used in the perception regularization is similar to the one from DCGAN (Radford et al., 2016). The detailed architecture is as follows:

Input: RGB image $x \in \mathbb{R}^{3\times64\times64}$
4×4 conv. $128ch$, ReLU, stride 2, padding 1
4×4 conv. $256ch$, ReLU, stride 2, padding 1
4×4 conv. $512ch$, ReLU, stride 2, padding 1
4×4 conv. $1024ch$, ReLU, stride 2, padding 1
4×4 conv. $128ch$, ReLU, stride 1, padding 0
Output: latent vector $z \in \mathbb{R}^{128}$

(a) Feature Map $\mathcal{F}$

Input: latent vector $z \in \mathbb{R}^{128}$
4×4 deconv. $128ch \rightarrow 1024ch$, ReLU, stride 1, padding 0
4×4 deconv. $1024ch \rightarrow 512ch$, ReLU, stride 2, padding 1
4×4 deconv. $512ch \rightarrow 256ch$, ReLU, stride 2, padding 1
4×4 deconv. $256ch \rightarrow 128ch$, ReLU, stride 2, padding 1
4×4 deconv. $128ch \rightarrow 3ch$, ReLU, stride 2, padding 1
Sigmoid
Output: RGB image reconstruction $x' \in \mathbb{R}^{3\times64\times64}$

(b) Decode $\mathcal{D}$

Input: latent vector $z \in \mathbb{R}^{128}$
Fully connected $128 \rightarrow 256$, ReLU,
Linear $256 \rightarrow 2$
Output: prediction $y \in \mathbb{R}^2$.

(c) Classifier $\mathcal{C}$

B.2 EXPERIMENTAL SETTING ON MNIST

For training the perceptual regularization, we perform 20 iterations of Adam optimizer (Kingma & Ba, 2015) with $\beta = (0.9, 0.999)$ and learning rate 10^{-3}. No data augmentation is used.

Input: Greyscale image $x \in \mathbb{R}^{1\times 28\times 28}$
4×4 conv. $16ch$, ReLU, stride 2, padding 1
4×4 conv. $16ch$, ReLU, stride 2, padding 1
4×4 conv. $32ch$, ReLU, stride 1, padding 0
4×4 conv. $32ch$, ReLU, stride 1, padding 0
Output: latent vector $z \in \mathbb{R}^{32}$

(d) Feature Map $\mathcal{F}$

Input: latent vector $z \in \mathbb{R}^{32}$
4×4 deconv. $32ch \to 32ch$, ReLU, stride 1, padding 0
4×4 deconv. $32ch \to 32ch$, ReLU, stride 1, padding 0
4×4 deconv. $32ch \to 16ch$, ReLU, stride 2, padding 1
4×4 deconv. $16ch \to 1ch$, ReLU, stride 2, padding 1
Sigmoid
Output: Greyscale image reconstruction $x' \in \mathbb{R}^{1\times 28\times 28}$

(e) Decode $\mathcal{D}$

Input: latent vector $z \in \mathbb{R}^{32}$
Fully connected $32 \to 64$, ReLU,
Linear $64 \to 10$
Output: prediction $y \in \mathbb{R}^{10}$.

(f) Classifier $\mathcal{C}$

B.3 EXPERIMENTAL SETTING ON SVHN

For the SVHN experiments, the inputs are RGB images of size 32×32. No data augmentation is used. Our architecture for the perception regularization is as follows:

Input: RGB image $x \in \mathbb{R}^{3\times32\times32}$
4×4 conv. $16ch$, ReLU, stride 2, padding 1
4×4 conv. $32ch$, ReLU, stride 2, padding 1
4×4 conv. $64ch$, ReLU, stride 2, padding 1
4×4 conv. $64ch$, ReLU, stride 1, padding 0
Output: latent vector $z \in \mathbb{R}^{64}$

(g) Feature Map $\mathcal{F}$

Input: latent vector $z \in \mathbb{R}^{64}$
4×4 deconv. $128ch \rightarrow 64$, ReLU, stride 1, padding 0
4×4 deconv. $64ch \rightarrow 32$, ReLU, stride 2, padding 1
4×4 deconv. $32ch \rightarrow 16$, ReLU, stride 2, padding 1
4×4 deconv. $16ch \rightarrow 3ch$, ReLU, stride 2, padding 1
Sigmoid
Output: RGB image reconstruction $x' \in \mathbb{R}^{3\times32\times32}$

(h) Decode $\mathcal{D}$

Input: latent vector $z \in \mathbb{R}^{64}$
Fully connected $64 \rightarrow 128$, ReLU,
Linear $128 \rightarrow 10$
Output: prediction $y \in \mathbb{R}^{10}$.

(i) Classifier $\mathcal{C}$