# Analyzing and Improving Representations with the Soft Nearest Neighbor Loss

**Nicholas Frosst** [1] **Nicolas Papernot** [1] **Geoffrey Hinton** [1]

## Abstract

We explore and expand the *Soft Nearest Neighbor Loss* to measure the *entanglement* of class manifolds in representation space: i.e., how close pairs of points from the same class are relative to pairs of points from different classes. We demonstrate several use cases of the loss. As an analytical tool, it provides insights into the evolution of class similarity structures during learning. Surprisingly, we find that *maximizing* the entanglement of representations of different classes in the hidden layers is beneficial for discrimination in the final layer, possibly because it encourages representations to identify class-independent similarity structures. Maximizing the soft nearest neighbor loss in the hidden layers leads not only to better-calibrated estimates of uncertainty on outlier data but also marginally improved generalization. Data that is not from the training distribution can be recognized by observing that in the hidden layers, it has fewer than the normal number of neighbors from the predicted class.

## 1. Introduction

From SVM kernels to hidden layers in neural nets, the similarity structure of representations plays a fundamental role in how well classifiers generalize from training data. Representations are also instrumental in enabling well-calibrated confidence estimates for model predictions. This is particularly important when the model is likely to be presented with outlier test data: *e.g.* to assist with medical diagnostics when a patient has an unknown condition, or more generally when safety or security are at stake.

In this paper, we use the labels of the data points to illuminate the class similarity structure of the internal representations learned by discriminative training. Our study of internal representations is structured around a loss function, the *soft nearest neighbor loss* (Salakhutdinov & Hinton, 2007), which we explore to measure the lack of separation of class manifolds in representation space—in other words, the *entanglement* of different classes. We expand upon the original loss by introducing a notion of temperature to control the perplexity at which entanglement is measured. We show several use cases of this loss including as an analytical tool for the progress of discriminative and generative training. It can also be used to measure the entanglement of synthetic and real data in generative tasks.

We focus mainly on the effect of deliberately *maximizing* the entanglement of hidden representations in a classifier. Surprisingly, we find that, unlike the penultimate layer, hidden layers that perform feature extraction benefit from being entangled. That is, they should *not* be forced to disentangle data from different classes. In practice, we promote the entanglement of hidden layers by adding our soft nearest neighbor loss as a bonus to the training objective. We find that this bonus regularizes the model by encouraging representations that are already similar to become more similar if they have different labels. The entangled representations form class-independent clusters which capture other kinds of similarity that is helpful for eventual discrimination.

In addition to this regularization effect, entangled representations support better estimates of uncertainty on outlier data, such as adversarial examples or inputs from a different distribution. In our empirical study, we measure uncertainty with the Deep k-Nearest Neighbors (DkNN): the approach relies on a nearest neighbor search in the representation spaces of the model to identify support in the training data for a given test input (Papernot & McDaniel, 2018). Since entangled representations exhibit a similarity structure that is less class-dependent, entangled models more coherently project outlier data that does not lie on the training manifold. In particular, data that is not from the training distribution has fewer than the normal number of neighbors in the predicted class. As a consequence, uncertainty estimates provided by the DkNN are better calibrated on entangled models.

The contributions of this paper are the following:

- We explore and expand the soft nearest neighbor loss to characterize the class similarity structure in representation space (Section 2). Informally, the loss measures

[1]Google Brain. Correspondence to: N. Frosst <frosst@google.com>, N. Papernot <papernot@google.com>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

how entangled class manifolds are and can be used to track progress in both discriminative and generative tasks (Section 3).

- We show that *maximizing* representation entanglement by adding a bonus proportional to the soft nearest neighbor loss to the training objective serves as a regularizer (Section 4).
- We find that entangled representations deal better with outlier data far from the training manifold, thus supporting better confidence estimates on adversarial examples or different test distributions (Section 5).

## 2. Soft Nearest Neighbor Loss

In the context of our work, the *entanglement* of class manifolds characterizes how close pairs of representations from the same class are, relative to pairs of representations from different classes. If we have very low entanglement, then every representation is closer to representations in the same class than it is to representations in different classes. In other words, a nearest neighbor classifier based on disentangled representations would have high accuracy.

The *soft nearest neighbor loss* (Salakhutdinov & Hinton, 2007) measures entanglement over labeled data. The loss computation can be approximated over a batch of data. Intuitively, we can think about this metric by imagining we are going to sample a neighboring point $j$ for every point $i$ in a batch, à la (Goldberger et al., 2005),[1] where the probability of sampling $j$ depends on the distance between points $i$ and $j$. The soft nearest neighbor loss is the negative log probability of sampling a neighboring point $j$ from the same class as $i$. Our definition introduces a new parameter, the temperature, to control the relative importance given to the distances between pairs of points.

**Definition.** The *soft nearest neighbor loss* at temperature $T$, for a batch of $b$ samples $(x, y)$, is:

$$l_{sn}(x, y, T) = -\frac{1}{b} \sum_{i \in 1..b} \log \left( \frac{\sum\limits_{\substack{j \in 1..b \\ j \neq i \\ y_i = y_j}} e^{-\frac{||x_i - x_j||^2}{T}}}{\sum\limits_{\substack{k \in 1..b \\ k \neq i}} e^{-\frac{||x_i - x_k||^2}{T}}} \right) \quad (1)$$

where $x$ may be either the raw input vector or its representation in some hidden layer. Temperature may be interpreted as the variance of Gaussians used to compute the probability of sampling neighboring points. At low temperatures, the loss is dominated by the small distances and the actual distances between widely separated representations are almost irrelevant. We open-sourced TensorFlow code outlining the matrix operations needed to compute this loss efficiently.

[1]The set of nearest neighbors for a given training point is also at the core of unsupervised techniques for nonlinear dimensionality reduction like locally-linear embeddings (Roweis & Saul, 2000).


*Figure 1.* A set of 200 2D points is sampled from a Gaussian and labeled randomly. Then, using gradient descent on the x and y coordinates of the points, the soft nearest neighbor loss is minimized to decrease entanglement. The 4 classes become more isolated. While a direct comparison with other losses like cross-entropy is not possible for this experiment, we inspect and compare non-entangled and entangled representation spaces later in the paper.

We plot different distributions annotated with their entanglement in Figure 1. As we minimize the soft nearest neighbor loss to decrease entanglement, the result is not necessarily each class collapsing to a *single* point. The loss is low when each point is closer to members of its own class than to other classes, but this can be achieved by having several widely separated pure cluster for each class. This is illustrated in Figure 13 (Appendix A) by introducing a second mode in each of the classes, which is preserved when entanglement is minimized by gradient descent on the soft nearest neighbor.

Like the triplet loss (Hoffer & Ailon, 2015), the soft nearest neighbor loss compares intra- to inter-class distances. However, a notable difference is that the triplet loss samples a single positive and negative point to estimate the separation of classes, whereas the soft nearest neighbor loss uses all positive and negative points in the batch. As visualized in Figure 2: when maximizing the soft nearest neighbor loss, this results in representations that are more spread out than the triplet loss. We show that this is a useful property of the soft nearest neighbor loss in Section 4 and defer a more complete treatment of the triplet loss to Appendix B.

**Temperature.** By varying the temperature $T$, it is possible to alter the value of the loss function significantly. As outlined in Equation 1, temperature divides the squared distance between points before it is negatively exponentiated.

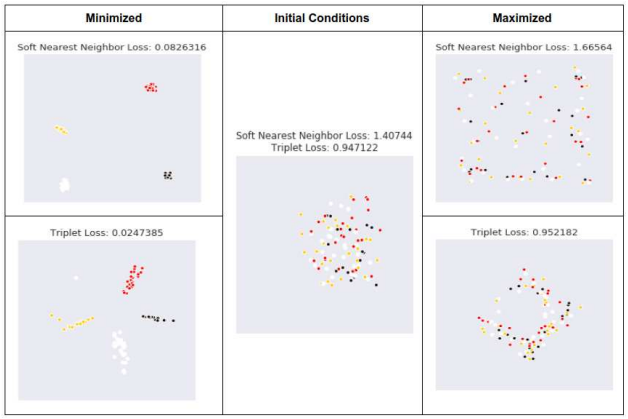


*Figure 2.* Comparing the triplet and soft nearest neighbor losses. The middle plot shows the initial condition (labels are reflected by color), the left plot the effect of minimizing either loss, and the right the effect of maximizing it.

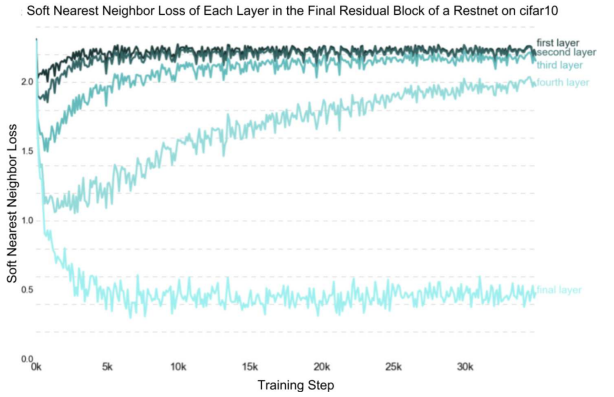


*Figure 3.* Entanglement of each layer within the last block of a ResNet on CIFAR-10, as measured with the soft nearest neighbor loss. Each layer initially disentangles data, but as training progresses and features are co-opted to represent sub features instead of classes, entanglement rises in all layers except for the final layer, which remains discriminative.

Thus, when temperature is large, the distances between widely separated points can influence the soft nearest neighbor loss. In the rest of this paper, we eliminate temperature as a hyperparameter by defining the entanglement loss as the minimum value over all temperatures:

$$l'_{sn}(x, y) = \arg\min_{T \in \mathbb{R}} l_{sn}(x, y, T) \tag{2}$$

We approximate this quantity by initializing $T$ to a predefined value and, at every calculation of the loss, optimizing with gradient descent over $T$ to minimize the loss.[2]

## 3. Measuring Entanglement during Learning

The soft nearest neighbor loss (SNNL) serves as an analytical tool to characterize the class similarity structure of representations throughout learning. In classifiers trained with cross-entropy, the SNNL illuminates how models learn to compose entangled layers for feature extraction with disentangled layers for classification. In generative models, the loss shows how well they learn to entangle the synthetic data with the real data from the distribution being modeled.

### 3.1. Discriminative Models

With the SNNL, we measure the entanglement of representations learned by each layer in the final block of a ResNet on CIFAR-10. In Figure 3, we distinguish two regimes. After an initial sharp decrease, the entanglement of lower layers of the block increases as training progresses. This suggests that the lower layers are discovering features shared by multiple classes. By contrast, the entanglement of the block's output layer consistently decreases throughout training because the last hidden layer must allow linear separation of the logit for the correct class from all the other logits.

Qualitatively consistent conclusions can be drawn at the granularity of blocks (rather than layer), as demonstrated by an experiment found in Appendix C. Later in Section 4, we build on this perhaps counter-intuitive finding and propose maximizing a soft nearest neighbor loss to regularize gradient descent on the cross-entropy loss.

### 3.2. Generative Models

We now turn to generative models, and verify that they eventually entangle synthetic data with real data. Then, we demonstrate how the soft nearest neighbor loss can act as an alternative to existing training objectives, in particular effectively replacing the discriminator used in GANs when semantics are captured by a distance in the input domain.

**Entanglement in GANs.** Synthetic data generated by GANs should be be highly entangled with real data because the generator is trained against a discriminator whose task is to discriminate between synthetic and real data (Goodfellow et al., 2014a). Here, we are no longer calculating the (self) entanglement of a training batch, but rather calculating the entanglement between a batch of real data and a batch of synthetic data. This comes down to applying the soft nearest neighbor loss on a data batch containing equal splits of real and synthetic points, labeled as 'real' or 'synthetic'.

In Figure 4, we report this measurement of entanglement at different stages of training a GAN on CIFAR10. We also visualize real and synthetic data using t-SNE (Maaten & Hinton, 2008). We observe that some modes of the input space are ignored by the generator, and conversely that some

[2] In practice, we found optimization to be more stable when we learn the inverse of the temperature.

modes of the generated space are not representative of the true data distribution. Note, however, how the real and synthetic data become less separable as training progresses, and how this is reflected in the entanglement score. This coherency between t-SNE and the soft nearest neighbor loss is to be expected given that both rely on similar calculations.

Similarly to the aforementioned use of the soft nearest neighbor loss as a metric to evaluate class manifold separation during classifier training, we measure entanglement between the real and generated data throughout training. In the context of generative models, there is only one soft nearest neighbor loss evaluation per architecture, because entanglement is only defined in the input domain. In Figure 6, we see that two variants of GANs exhibit different regimes of entanglement between synthetic and real CIFAR10 data as training progresses. We repeat the experiment on MNIST with qualitatively identical results in the Appendix D.

**Soft nearest neighbor loss as an objective.** Given that GANs implicitly maximize entanglement, it is natural to ask whether the soft nearest neighbor loss can be used directly as a training objective for the generator. To test this hypothesis, we replaced the discriminator (and its loss) with an inverse soft nearest neighbor loss in the GAN implementation used in our previous experiments on MNIST: i.e., the generator is now encouraged to maximize entanglement computed over a batch of real and synthetic data directly in pixel space.

On MNIST, this results in realistic and varied generated images (see Figure 6), which include all classes. Modes of the classes are captured by the generator, with for instance both the curly and straight "2". They are however noticeably smoother than data generated by traditional GANs. As a possible explanation, the generator maximizes the soft nearest neighbor loss evaluated on a batch when its output lies in between two training examples. However, this strategy does not generalize to more complex datasets like CIFAR10, most likely because the Euclidean distance in pixel space used in the soft nearest neighbor loss does not adequately capture the underlying semantics of images.

This limitation may most likely be overcome by measuring entanglement in a learned space, instead of pixel space. A potential preliminary instantiation of this intuition is explored in Appendix I: we replace the cross-entropy loss that a normal discriminator minimizes with the soft nearest neighbor loss applied to a learned space. In this way, the discriminator learns a projection of the real and synthetic data that separates one from the other.

Our proof-of-concept from Appendix I demonstrates that this strategy succeeds on MNIST. This may also overcome the previously mentioned limitations for CIFAR10 image generation. However, our focus being classification, we leave a comprehensive investigation of the interplay between entanglement and generative modeling as future work.

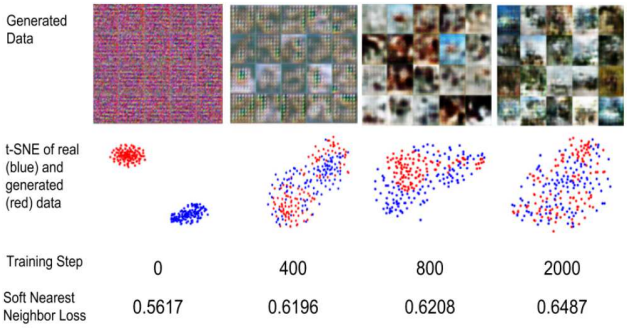


*Figure 4.* As training of vanilla GANs progresses, here on CIFAR10, the generator learns to entangle synthetic and training data, as confirmed by their increasing overlap in the t-SNE visualization as well as the larger SNNL values.

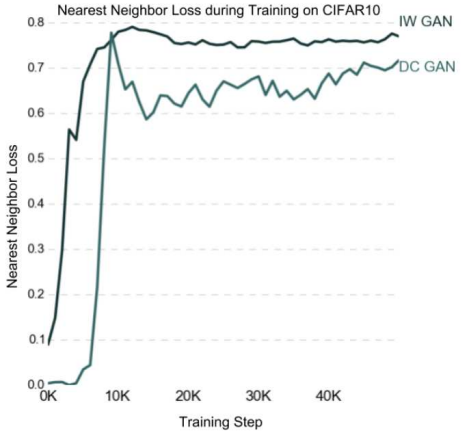


*Figure 5.* Entanglement of real and synthetic (generated) data throughout training, as measured with the SNNL on two types of GAN architectures trained on CIFAR10.

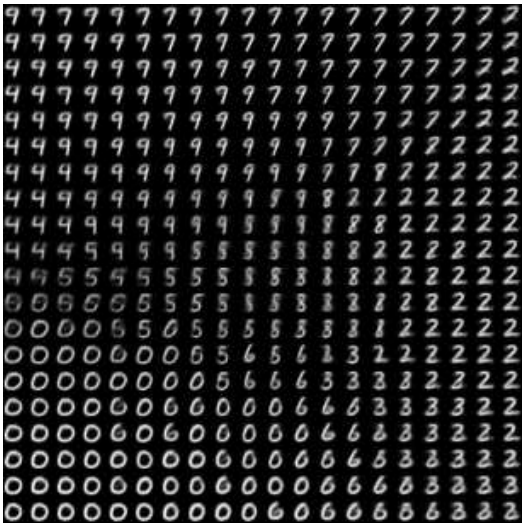

*Figure 6.* Samples from a generative model trained to maximize the SNNL between synthetic and training data. The grid extrapolates over 2 dimensions of the input noise.

# 4. Entangling Representation Spaces

Apart from its characterization of similarity in representation spaces, we found that the soft nearest neighbor loss may also serve as a training objective for generative models. At first, it appears that for discriminative models, one should encourage lower entanglement of internal representations by minimizing the SNNL. Indeed, this would translate to larger margins between different classes (Elsayed et al., 2018).

However, we show here that *maximizing* entanglement—in addition to minimizing cross-entropy—regularizes learning. Specifically, training a network to minimize cross-entropy and maximize the soft nearest neighbor loss reduces overfitting and achieves marginally better test performance. In Section 5, we will furthermore show that it promotes a class similarity structure in the hidden layers that better separates in-distribution from out-of-distribution data.

## 4.1. Intuition behind Maximizing Entanglement

Clustering data based on its labels is a natural avenue for learning representations that discriminate: once a test point is assigned to a cluster of training points, its label can be inferred. This is referred to as the cluster assumption in the semi-supervised learning literature (Chapelle et al., 2009). However if test data is not represented in one of these class-homogeneous clusters, the behaviour of the network and the subsequent predicted label may be inconsistent. We argue that projecting all points in a class to a homogeneous clusters can be harmful to generalization and robustness.

Instead, we propose regularizing models by maximizing entanglement through the SNNL to develop class-independent similarity structures. This not-only promotes spread-out intraclass representations, but also turns out to be good for recognizing data that is not from the training distribution by observing that in the hidden layers, such data has fewer than the normal number of neighbors from the predicted class.

Concretely, we minimize an objective that balances a cross-entropy term on logits and a soft nearest neighbor term on each hidden representation with a hyper-parameter $\alpha < 0$, we represent the network as a series of transformations $f^k$, where $f^1$ is the first layer and $f^k$ is the logit layer.

$$l(f, x, y) = -\sum_j y_j \log f^k(x_j) + \alpha \cdot \sum_{i \in k-1} l'_{sn}(f^i(x), y) \tag{3}$$

This may seem counter-intuitive but we note that many regularizers take on the form of two seemingly mutually exclusive objectives. For example label smoothing (Pereyra et al., 2017) can be thought of trying to train a network to make accurate and confident predictions, but not overly confident. Similarly, dropout prompts individual neurons to operate independently from other—randomly deactivated—neurons, while still learning features that can be meaningfully combined (Srivastava et al., 2014). Here, our training objective simultaneously minimizes cross-entropy and maximizes the soft nearest neighbor loss. In other words, the model is constrained to learn representations whose similarity structure facilitates classification (separability) but also entanglement of representations from different classes (inseparability).

## 4.2. Soft Nearest Neighbor Loss as a Regularizer

We first measure the generalization of models that maximize the soft nearest neighbor loss in addition to minimizing cross-entropy. We trained a convolutional network[3] on MNIST, Fashion-MNIST and SVHN, as well as a ResNet[4] on CIFAR10. Two variants of each model were trained with a different objective: (1) a *baseline* with cross-entropy only and (2) an *entangled* variant balancing both cross-entropy and the soft nearest neighbor loss as per Equation 3. As reported in Table 1, all entangled models marginally outperformed their non-entangled counterparts to some extent.

To validate that maximizing entanglement is beneficial for generalization, we fine-tuned the hyperparameter $\alpha$ that balances the cross-entropy and soft nearest neighbor terms in our objective. The search was conducted on our CIFAR10 model using a strategy based on Batched Gaussian Process Bandits (Desautels et al., 2014). Because both positive and negative values of $\alpha$ were considered, this search explored respectively both minimization and maximization of representation entanglement. As illustrated by Figure 7, the search independently confirmed that maximizing entanglement led to better test performance as it eventually converged to large negative values of $\alpha$.

To explain the increased test performance of entangled models, we hypothesized that the entanglement term added to our training objective serves as a regularizer. To verify this, we measured the cross-entropy loss on training and test data while training the non-entangled and entangled variants of our models for a large number of steps. This allowed for overfitting. We draw the corresponding learning curves for SVHN in Figure 8 and observe that the entangled model not only overfits at a later stage in training (about 5,000 steps later), it also overfits to a much lesser degree.

However, we stress that the goal is not to demonstrate that the SNNL can outperform existing regualizers but rather to show that the benefits it brings in terms of representation geometry do not come at the cost of decreased generalization.

[3]The architecture we used was made up of two convolutional layers followed by three fully connected layers and a final softmax layer. The network was trained with Adam at a learning rate of 1e-4, a batch size of 256 for 14000 steps.

[4]The ResNet v2 with 15 layers was trained for 106 epochs with a exponential decreasing learning rate starting at 0.4.

| CNN Model | Test Accuracy | Entangled | Baseline |
|---|---|---|---|
| MNIST | Best | **99.23%** | 98.83% |
| | Average | **99.16%** | 98.82% |
| Fashion-MNIST | Best | **91.48%** | 90.42% |
| | Average | **91.06%** | 90.25% |
| SVHN | Best | **88.81%** | 87.63% |
| | Average | **89.90%** | 89.71% |
| **ResNet Model** | **Test Accuracy** | **Entangled** | **Baseline** |
| CIFAR10 | Best | **91.220%** | 90.780% |
| | Average | **89.900%** | 89.713% |

*Table 1.* Using a composite loss, to minimize cross-entropy loss and maximize entanglement through the SNNL, marginally increases test performance. Values are averaged over 4 runs for the CNN and 100 runs for the ResNet. While we note that baseline accuracies we report are below state-of-the-art, this is an intentional experimental design choice. Indeed, we wanted to isolate the behavior of our soft nearest neighbor loss from other factors (e.g., dropout or other regularizers) that may impact representation spaces.

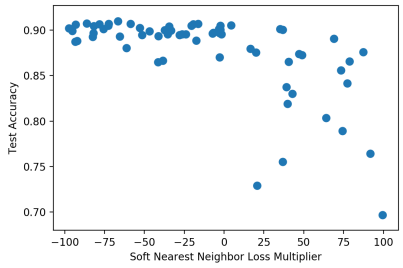


*Figure 7.* Test accuracy as a function of the soft nearest neighbor hyper-parameter $\alpha$ for 64 training runs of a ResNet on CIFAR10. Each run is selected by Vizier (Golovin et al., 2017) to maximize validation accuracy by tuning the learning rate, SNNL hyper-parameter $\alpha$, and temperature $T$.

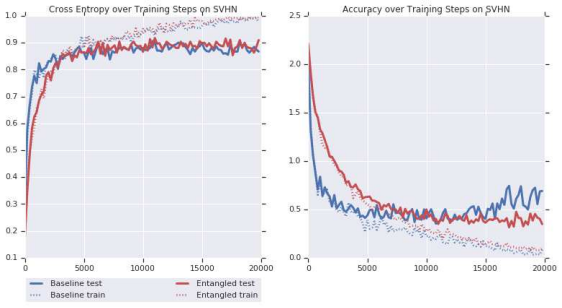


*Figure 8.* Accuracy and cross-entropy for baseline (blue) and entangled (red) models as a function of the number of training steps. In addition to increased test accuracy (left), the smaller gap between cross-entropy on training and test data (right) for entangled models illustrates how they begin to overfit later and to a lesser degree than non-entangled models. Curves are averaged over two runs for both models.

## 5. Entangled Models in Adversarial Settings

Given the improved—more class-independent—similarity structure of entangled representations obtained through maximizing the soft nearest neighbor loss, we hypothesize that entangled models also offer better estimates of their uncertainty. Here, we do *not* claim robustness to adversarial examples but rather show that entangled representations help distinguish outliers from real data. We validate this by considering two types of out-of-distribution test data: first, maliciously-crafted adversarial examples, and second, real inputs from a different test distribution. We find that hidden layers of entangled models consistently represent outlier data far away from the expected distribution's manifold.

It is natural to ask if reduced class margins make entangled representations more vulnerable to adversarial perturbations. This is not necessarily the case. In fact, we show in Appendix E that models with state-of-the-art robustness on MNIST have higher entanglement than non-robust counterparts. Furthermore, recent work has found that when models concentrate data, they are more vulnerable to adversarial examples (Mahloujifar et al., 2018), whereas entangled models encourage intraclass clusters to spread out.

**Attack techniques.** Our study considers both white-box and black-box threat models. Given access to gradients in the white-box setting, various heuristics and optimization algorithms allow the adversary to create adversarial examples (Biggio et al., 2013; Szegedy et al., 2013). Here, we use both single-step and iterative attacks: the Fast Gradient Sign Method (Goodfellow et al., 2014b) and Basic Iterative Method (Kurakin et al., 2016). When gradients are unavailable, as is the case for black-box interactions (i.e., the adversary only has access to the label predicted), a common strategy is to first find adversarial examples on a substitute model and then transfer them to the victim model (Szegedy et al., 2013; Papernot et al., 2017). Adversarial perturbations are said to be *universal* if they change a model's prediction into a chosen class once added to *any* input (Goodfellow et al., 2014b; Moosavi-Dezfooli et al., 2017).

**Uncertainty estimation.** Estimating the epistemic uncertainty that stems from the finite nature of datasets analyzed by models during learning remains an open problem. In our work, we apply a recent proposal called the Deep k-Nearest Neighbors (Papernot & McDaniel, 2018) that computes the credibility of each test-time prediction; a metric that reflects how well the training data supports this prediction. The approach consists in running a k-nearest neighbors search in the representation space learned by each hidden layer so as to extract the k training points whose representation is closest to the predicted representation of the test point considered. If the labels of these nearest training points largely agree with the test label being predicted, the prediction is

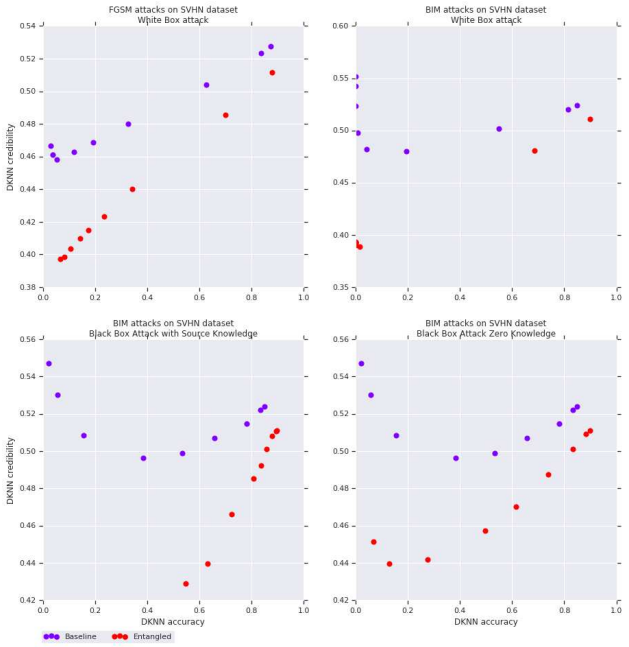


*Figure 9.* DkNN credibility (i.e., uncertainty estimate) as a function of accuracy on SVHN (averaged over two runs). Each point corresponds to adversarial examples generated with $\varepsilon \in [0.01, 0.5]$. Iterative attacks use 1000 steps and $\alpha = 0.01$ for all $\varepsilon$. Plots are shown for white-box FGSM attack (top left), white-box BIM attack (top right), black-box attacks with source knowledge (bottom left), black-box attacks with zero knowledge (bottom right). Source knowledge implies the adversary is aware of the defense and transfers adversarial examples from a model trained with the same loss, whereas zero knowledge adversaries always transfer from a model trained with cross-entropy. This allows us to rule out most common forms of gradient masking. In all cases, entangled models yield credibility estimates that are more correlated with accuracy, and the two bottom graphs show that they suffer less from transferability.

assigned high credibility. Otherwise, it is assigned a low credibility score, which implies it should not be relied upon. A holdout dataset is used to calibrate the expected level of agreement between the training and test data.

### 5.1. Entangled Representations support more Calibrated DkNN Estimates of Uncertainty

In the original proposal, the DkNN is applied to vanilla neural networks without modifying the way they are trained. Intuitively, training with the soft nearest neighbor loss should impact the credibility predicted by the DkNN because it modifies the class similarity structure of hidden representations that are core to the analysis performed by the DkNN.

Using MNIST, Fashion-MNIST and SVHN, we compare two models : one trained with cross-entropy only and one with the composite loss from Equation 3 that includes a cross-entropy term and soft nearest neighbor term. We compare how the two models' credibility estimates correlate with their predictive accuracy. Ideally, the relationship between the two should be the identity; if a DkNN system was perfectly calibrated then inputs that were correctly classified would have 100% credibility while inputs that were incorrectly classified would have 0% credibility.

We tested each model on FGSM and BIM adversarial examples assuming white-box access to the model, with progressively larger perturbations ($\varepsilon$ gradient step). We also considered adversarial examples crafted with the BIM attack but transferred from a different model. This black-box attack enables us to test for gradient masking. In Figure 9, we then plotted the average DkNN credibility (low credibility corresponds to higher uncertainty) with respect to the classification accuracy. Each point corresponds to a different perturbation magnitude. While the credibility is not perfectly linear with respect to the accuracy for either the standard or entangled model, the correlation between credibility and accuracy is consistently higher for entangled models in both the white-box and black-box settings.

To explain this, we t-SNE representations in Appendix H and find that entangled models better separate adversarial data from real data in activation space. This in turn implies that adversarial data can be recognized as not being part of the distribution by observing that it has fewer than the normal number of neighbors from the predicted class.

### 5.2. Transferability and Representation Entanglement

Transferability—the fact that adversarial examples for one model are also often misclassified by a different model—was empirically found to apply to a wide range of model pairs, despite these models being trained with different ML techniques (e.g., decision trees and neural nets) or subsets of data. Several hypotheses were put forward to explain why this property holds in practice, including gradient alignment.

This is visualized in Figure 10, which plots gradients followed by a targeted FGSM attack in 2D using t-SNE. The plot stacks the visualizations for two different models. Coherent clusters exist across the two individual models: gradients that are adversarial to one model are likely to be aligned with gradients that are adversarial to the second model.

However, this gradient alignment does not hold in entangled models. When we repeat the same experiment with a standard cross-entropy model and an entangled model, or two entangled models, the clusters are no longer coherent across pairs of models—as illustrated in Figure 11. This suggests that while adversarial examples can still be found in the white-box setting by following the gradients of a specific entangled model, it is harder to find perturbations that are universal (i.e., apply to any test input) or transferable

(i.e., apply across different entangled models) simply by exploiting gradient alignment across inputs or models.

### 5.3. Out-of-Distribution Test Inputs

Unlike techniques like adversarial training (Szegedy et al., 2013), training with the soft nearest neighbor loss relies only on the original training data and makes no assumptions about a particular algorithm used to generate the out-of-distribution examples. Hence, having shown that training a network to maximize entanglement leads to representations that better separate adversarial data from real data, we expect this behaviour to be consistent across any data sampled from something other than the expected test distribution. This includes inputs from a different test distribution.

To test this we can train a network on SVHN and see what its behavior is like on CIFAR10: test examples from CIFAR10 should be represented very differently from the SVHN test examples. This is indeed what we observe in Figure 12, which uses t-SNE to visualize how the logits represent SVHN and CIFAR10 test inputs when a model is trained with cross-entropy only or with the soft nearest neighbor loss to maximize entanglement. The vanilla model makes confident predictions in the SVHN classes for the CIFAR10 inputs (because they are represented close to one another), whereas the entangled model separates all of the CIFAR10 data in a distinct cluster and preserves the SVHN clusters. A similar experiment on a MNIST model using notMNIST as out-of-distribution test inputs is found in Appendix G.

## 6. Conclusions

We expanded on and explored novel use cases of the soft nearest neighbor loss. It can serve as a tool to characterize the class similarity structure of representations, allowing us to measure learning progression of discriminative models. The loss also captures how generative models entangle synthetic and real data, and can thus serve as a generative loss itself. Furthermore, by adding the loss as a bonus to a classifier's training objective, we are able to boost test performance and generalization. Because entangled representations are encouraged to spread out data further in activation space (see Figure 13), they represent outlier data more consistently apart from real data (see Figure 22). This in turn means outlier data is easily rejected by observing that it is supported by fewer neighbors from the predicted class, as captured by our improved uncertainty estimates.

ACKNOWLEDGMENTS

We would like to thank Martin Abadi, Samy Bengio, Nicholas Carlini, Yann Dauphin, Ulfar Erlingsson, Danijar Hadner, Ilya Mironov, Sara Sabour, Kunal Talkwar, Nithum Thain. We also thank the anonymous reviewers.

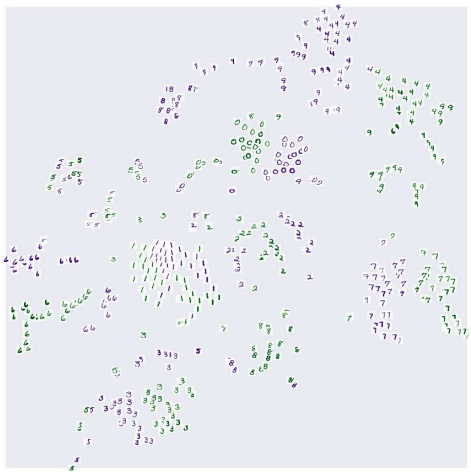

*Figure 10.* t-SNE visualization of gradients computed by a FGSM attack targeting class 1 on two vanilla models, one in green the other in blue.

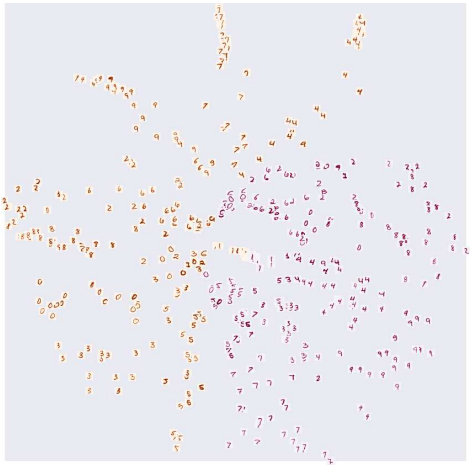

*Figure 11.* t-SNE visualization of gradients computed by a FGSM attack targeting class 1 on two entangled models, one in red the other in orange.

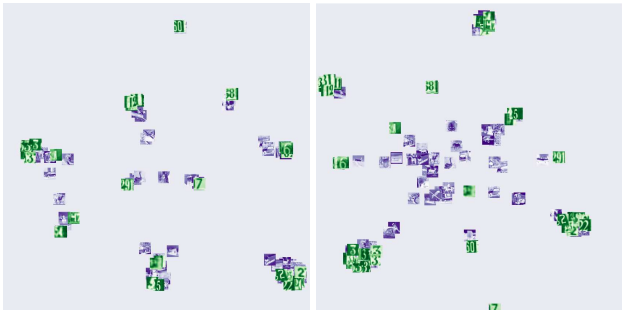

*Figure 12.* t-SNE of logits for in-distribution (SVHN—green) and out-of-distribution (cifar10—blue) test data learned by a baseline (left) and entangled (right) model. Out of distribution data is easier to separate from the true data for the entangled model than it is for the baseline model.

## References


Azadi, S., Feng, J., Jegelka, S., and Darrell, T. Auxiliary image regularization for deep cnns with noisy labels. *arXiv preprint arXiv:1511.07069*, 2015.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *The Journal of Machine Learning Research*, 15(1):3873–3923, 2014.

Elsayed, G. F., Krishnan, D., Mobahi, H., Regan, K., and Bengio, S. Large margin deep networks for classification. *arXiv preprint arXiv:1803.05598*, 2018.

Goldberger, J., Hinton, G. E., Roweis, S. T., and Salakhutdinov, R. R. Neighbourhood components analysis. In *Advances in neural information processing systems*, pp. 513–520, 2005.

Golovin, D., Solnik, B., Moitra, S., Kochanski, G., Karro, J., and Sculley, D. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1487–1495. ACM, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014a.

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

Hoffer, E. and Ailon, N. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pp. 84–92. Springer, 2015.

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

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

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.

Mahloujifar, S., Diochnos, D. I., and Mahmoody, M. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. *arXiv preprint arXiv:1809.03063*, 2018.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. *arXiv preprint*, 2017.

Papernot, N. and McDaniel, P. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

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.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326, 2000.

Salakhutdinov, R. and Hinton, G. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pp. 412–419, 2007.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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.