Learning Transformation Invariant Representations with Weak Supervision

Benjamin Coors[1,2], Alexandru Condurache[2], Alfred Mertins[3] and Andreas Geiger[1,4]
[1] *Autonomous Vision Group, MPI for Intelligent Systems, Tübingen, Germany*
[2] *Robert Bosch GmbH, Leonberg, Germany*
[3] *Institute for Signal Processing, University of Lübeck, Germany*
[4] *Computer Vision and Geometry Group, ETH Zürich, Switzerland*

Keywords: Deep Learning, Transformation Invariance, Weak Supervision, Object Recognition.

Abstract: Deep convolutional neural networks are the current state-of-the-art solution to many computer vision tasks. However, their ability to handle large global and local image transformations is limited. Consequently, extensive data augmentation is often utilized to incorporate prior knowledge about desired invariances to geometric transformations such as rotations or scale changes. In this work, we combine data augmentation with an unsupervised loss which enforces similarity between the predictions of augmented copies of an input sample. Our loss acts as an effective regularizer which facilitates the learning of transformation invariant representations. We investigate the effectiveness of the proposed similarity loss on rotated MNIST and the German Traffic Sign Recognition Benchmark (GTSRB) in the context of different classification models including ladder networks. Our experiments demonstrate improvements with respect to the standard data augmentation approach for supervised and semi-supervised learning tasks, in particular in the presence of little annotated data. In addition, we analyze the performance of the proposed approach with respect to its hyperparameters, including the strength of the regularization as well as the layer where representation similarity is enforced.

1 INTRODUCTION

A central problem in computer vision is to train classifiers which are robust to geometric transformations of the input that are irrelevant to the problem at hand. The most commonly used solution to ensure robustness of a classifier to geometric transformations is data augmentation (Simard et al., 2003; Krizhevsky et al., 2012; Laptev et al., 2016). Data augmentation artificially enlarges the training set and acts as a regularizer, which prevents a classifier from overfitting to the training set. As an alternative to data augmentation, transformation invariances can be directly encoded into the convolutional filters of convolutional neural networks (CNNs) (Cohen and Welling, 2016; Worrall et al., 2017; Zhou et al., 2017). However, these approaches are currently limited to simple geometric transformations such as rotations.

In this work, we propose to leverage an unsupervised similarity loss for training deep neural networks invariant to arbitrary transformations. The similarity loss is computed with respect to transformed copies of an input and presents a very simple and effective regularizer, enforcing the desired transformation invariances. In contrast to naïve data augmentation, it encourages smooth decision boundaries with respect to transformations of the input and leads to higher performance, in particular in the presence of little annotated examples. Besides, our method allows for easy incorporation of additional unlabeled examples, as the similarity loss does not utilize label information and is thus suitable for semi-supervised learning tasks. To the best of our knowledge, this is the first work to propose a similarity loss for training transformation invariant CNNs with weak supervision. The contributions of this paper are:

- We propose a similarity loss which acts as an additional regularizer and utilizes unlabeled training data for learning transformation invariance.
- We present a detailed investigation on the weighting and placement of the loss.
- We show improved performance in supervised and semi-supervised learning on rotated MNIST and GTSRB when little labeled data is available.

Coors, B., Condurache, A., Mertins, A. and Geiger, A.
Learning Transformation Invariant Representations with Weak Supervision.
DOI: 10.5220/0006549000640072
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 5: VISAPP*, pages 64-72
ISBN: 978-989-758-290-5

2 RELATED WORK

As an alternative to data augmentation, knowledge about geometric transformations can be directly encoded into the filters of a convolutional neural network. Scattering convolution networks use predefined wavelet filters to create networks that are invariant to translations, rotations, scaling and deformations (Bruna and Mallat, 2013; Sifre and Mallat, 2013). While scattering networks guarantee stability to geometric transformations, their parameters cannot be trained and thus they are generally outperformed by supervised deep convolutional networks (Oyallon and Mallat, 2015).

Consequently, several recent works have suggested to combine the encoding of invariances with the learning of the convolutional filters (Kivinen and Williams, 2011; Sohn and Lee, 2012; Cohen and Welling, 2016; Worrall et al., 2017; Zhou et al., 2017). Transformation invariant or equivariant restricted Boltzmann machines infer the best matching filters by transforming them using linear transformations (Kivinen and Williams, 2011; Sohn and Lee, 2012). Similarly, group equivariant CNNs apply learned base filters under different transformations and pool their responses to create invariant representations (Cohen and Welling, 2016). Harmonic networks exhibit global rotation equivariance (Worrall et al., 2017). While these works have demonstrated state-of-the-art results and shown promise in improving the data-efficiency of deep convolutional networks, they are, unlike our approach, typically restricted to simple transformations (*e.g*, rotations).

Another approach to transformation invariance in CNNs is to resample the input space. An example for this approach are spatial transformer networks (STNs) (Jaderberg et al., 2015), which use a separate network to learn the parameters of a spatial transformation of an input. Based on the predicted transformation parameters a sampling grid is created and applied to the input. A more lightweight alternative to STNs are deformable convolutional networks (Dai et al., 2017), which do not learn transformation parameters or warp the feature map but instead directly learn offsets to the regular sampling grid of standard convolutions. While these approaches increase the flexibility of neural networks in handling geometric transformations, they assume that a canonical representation can be easily deduced from the input.

Because of the shortcomings of the aforementioned techniques, data augmentation remains one of the most commonly used solutions for making deep networks invariant to complex input transformations (Simard et al., 2003; Krizhevsky et al., 2012). A recently proposed variant of data augmentation is transformation-invariant pooling (TI-pooling) (Laptev et al., 2016), which feeds multiple augmented copies of an input into the network and pools their responses. While this simple idea works well in practice, test time complexity grows exponentially with the dimension of the transformation, rendering this approach infeasible for real-time applications. In contrast, our loss encourages representation similarity during training and does not affect test time performance.

The proposed approach is also related to the topic of self-supervised learning, where freely available auxiliary labels are used to train algorithms without human supervision. Recently proposed proxy tasks for self-supervision include context prediction (Doersch et al., 2015), solving jigsaw puzzles (Noroozi and Favaro, 2016) or predicting egomotion signals (Agrawal et al., 2015). Compared to these works on self-supervision, our work does not use an auxiliary training loss but directly optimizes the desired loss metric.

The idea of using an input sample more than once in each training step has previously been proposed in the context of protecting neural networks against adversarial perturbations (Zheng et al., 2016; Miyato et al., 2016). Similar to the idea of improving model generalization by injecting adversarial examples during training, we aim to improve model generalization w.r.t. transformations by enforcing similarity between feature representations of transformed input images.

Besides, state-of-the-art results in semi-supervised learning have been recently presented using a similarity loss in combination with a mutual-exclusivity loss (Sajjadi et al., 2016) or via an imaginary walker that is tasked with forming ”associations” between embeddings (Haeusser et al., 2017). In contrast to these works, here we investigate the utility of a similarity loss when learning representations *invariant to geometric transformations* and present a detailed analysis on the placement and weighing of the loss. Our loss applies to the supervised and semi-supervised setting and is particularly effective in the presence of little labeled data.

3 METHOD

While the proposed similarity loss is applicable to a variety of tasks, we use image classification as a test bench in this work. Let $x \in \mathbb{R}^{w \times h \times c}$ denote an image of dimensions $w \times h$ with c channels and let $f : \mathbb{R}^{w \times h \times c} \to \mathbb{R}^{C}$ be a non-linear mapping represented by a neural network which takes an input image x and produces a score for each of the C classes. Let further $t_0, t_1 \in \mathcal{T}$ denote two transformations from a set of transformations $\mathcal{T}$ (*e.g*, rotation, affine, perspective) which take the input image x and produce transformed versions $t_0(x) \in \mathbb{R}^{w\times h\times c}$ and $t_1(x) \in \mathbb{R}^{w\times h\times c}$ of it. Finally, let $f_l(t(x))$ denote the feature maps of the neural network in layer l when passing $t(x)$ as input. For clarity, we will drop the dependency on the input image x in the following.

In order to encourage a neural network to learn transformation invariant representations, we propose the use of a similarity loss $\mathcal{L}_{sim}$ which penalizes large distances between the predictions or feature embeddings of transformed copies of the input. The similarity loss is computed using a siamese network architecture, where the transformed copies of the input are simultaneously fed into separate streams of the network that share their weights. By transforming both inputs, convergence of the model is accelerated and overfitting to small label sets is avoided. An abstract network architecture, where the similarity loss is applied at the final layer L, is illustrated in Figure 1(a).

At inference time only a single stream of the network is used, keeping runtime constant with respect to the size of the transformation space.

The similarity loss $\mathcal{L}_{sim}$ is added to the supervised classification loss $\mathcal{L}_c$, which is applied on the output of both network streams, to form the total loss $\mathcal{L}_{total}$ for a data point where a weight parameter λ controls the influence of $\mathcal{L}_{sim}$:

$$\mathcal{L}_{total} = \mathcal{L}_c + \lambda\mathcal{L}_{sim} \tag{1}$$

Here, $\mathcal{L}_c$ is the usual cross-entropy loss applied to the softmax outputs $\sigma(f_L(t_0))$ and $\sigma(f_L(t_1))$ of the final network layer L for the transformed input sample:

$$\mathcal{L}_c = -\sum_{i=1}^{C} y_i \log\sigma_i(f_L(t_0)) - \sum_{i=1}^{C} y_i \log\sigma_i(f_L(t_1)) \tag{2}$$

where $\sigma_i(x) = \exp(x_i)/\sum_{j=1}^{C}\exp(x_j)$ denotes the softmax function and $y_i = 1$ if i is the ground truth class and $y_i = 0$ otherwise.

The similarity loss encourages the output of the neural network at layer l, f_l, to be similar for both streams. It is defined as the distance between the outputs of the transformed input pair at layer l for an appropriate distance metric $\mathcal{D}(\cdot,\cdot)$:

$$\mathcal{L}_{sim} = \mathcal{D}(f_l(t_0), f_l(t_1)) \tag{3}$$

We propose to use a distance metric which measures the correspondence between the likelihood of the transformed input copies. More specifically, $\mathcal{D}(\cdot,\cdot)$ is calculated by flattening the network output f_l at a given layer l and applying the softmax activation:

$$\mathcal{D}(f_l(t_0), f_l(t_1)) = -\sum_{i=1}^{C} \sigma_i(f_l(t_0)) \log\sigma_i(f_l(t_1)) \tag{4}$$

A similar distance metric has previously been proposed by Zheng et al. (2016) in order to stabilize models against small input perturbations such as the addition of uncorrelated Gaussian noise. It is inspired by the work of Miyato et al. (2016) on virtual adversarial training, which showed that a distance function based on the Kullback-Leibler (KL) divergence smoothens the model distribution with respect to the input around each data point. Our work extends this approach to training models for invariance to geometric transformations of the input and is the first to perform a detailed investigation on the optimal weighing and placement of the loss.

Since the similarity loss $\mathcal{L}_{sim}$ does not require label information, it enables semi-supervised learning with partially labeled data. Until recently, ladder networks were the state-of-the-art architecture for semi-supervised learning (Rasmus et al., 2015). Ladder networks are denoising autoencoders with lateral connections, into which the similarity loss $\mathcal{L}_{sim}$ can be easily integrated by duplicating the corrupted or uncorrupted encoder path of the ladder network. The duplicated encoder path again shares its weight with the other encoder paths of the ladder network. A simple ladder network architecture where the corrupted encoder path of the ladder network is duplicated and $\mathcal{L}_{sim}$ is applied on the final output layer L is illustrated in Figure 1(b).

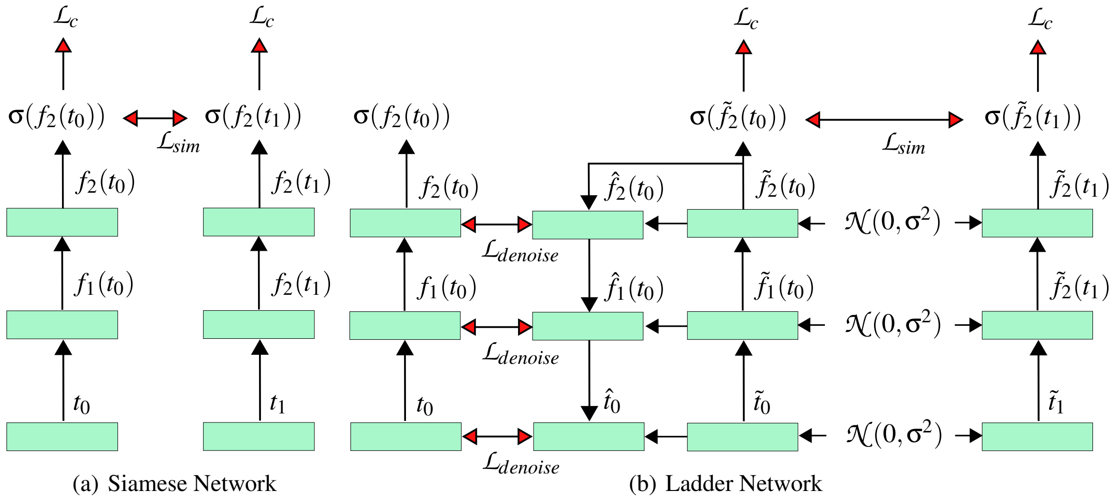
As before, $\mathcal{L}_{sim}$ is added to the total loss $\mathcal{L}_{total}$ where it serves as a second unsupervised loss next to the denoising loss $\mathcal{L}_{denoise}$, which aims to minimize the difference between a clean layer output f_l and the output of a denoising function $\hat{f}_l$ given a corrupted output $\tilde{f}_l$ on all L layers of the network. The classification loss $\mathcal{L}_c$ is then computed on the outputs of the noisy encoder paths $\tilde{f}_L$ where α is a weight parameter of the denoising loss $\mathcal{L}_{denoise}$ and $f_0(t_0) = t_0$.

$$\mathcal{L}_{total} = \mathcal{L}_c + \alpha\mathcal{L}_{denoise} + \lambda\mathcal{L}_{sim} \tag{5}$$

$$\mathcal{L}_c = -\sum_{i=1}^{C} y_i \log\sigma_i(\tilde{f}_L(t_0)) - \sum_{i=1}^{C} y_i \log\sigma_i(\tilde{f}_L(t_1)) \tag{6}$$

$$\mathcal{L}_{denoise} = \sum_{l=0}^{L} ||f_l(t_0) - \hat{f}_l(t_0)||^2 \tag{7}$$

In all models, the classification loss $\mathcal{L}_c$ is only applied on the labeled training samples. On the other hand, the similarity loss $\mathcal{L}_{sim}$ and, in case of a ladder network, the denoising loss $\mathcal{L}_{denoise}$ can be applied on both labeled and unlabeled samples of the training data.



(a) Siamese Network

(b) Ladder Network

Figure 1: Abstract network architectures considered in this work. A similarity loss $\mathcal{L}_{sim}$ is placed on the the final layer of the network in order to enforce similarity between outputs of transformed copies of an input.

4 EXPERIMENTS

We first validate our approach in terms of learning *rotation invariant* representations on the classical rotated MNIST task (Larochelle et al., 2007). Second, we demonstrate the effectiveness of our technique on the more challenging German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2012). Incorporating *perspective invariances* using the proposed similarity loss, our method leads to significant improvements over the baselines for this task.

4.1 Experimental Setup

The rotated MNIST classification task (Larochelle et al., 2007) is the standard benchmark for evaluating transformation invariance in neural networks (Sohn and Lee, 2012; Cohen and Welling, 2016; Laptev et al., 2016; Worrall et al., 2017), despite possible ambiguities between rotated digits such as a rotated 6 and 9. The rotated MNIST dataset (see Figure 2) was created by rotating MNIST digits with uniformly sampled angles between 0 and 2π radians and consists of $12,000$ training and $50,000$ test samples. As in the original MNIST dataset (Lecun et al., 1998), the images are greyscale and of size 28×28 pixels. We split the dataset into $10,000$ training and $2,000$ validation samples for determining the hyperparameter λ.

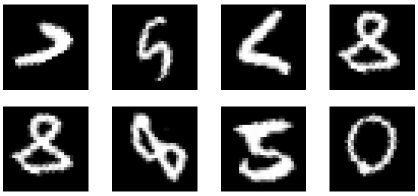


Figure 2: Example images from rotated MNIST (Larochelle et al., 2007).

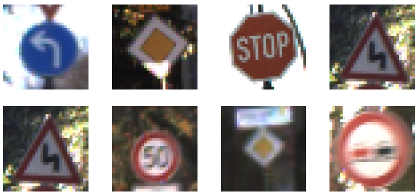


Figure 3: Example images from GTSRB (Stallkamp et al., 2012).

The German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2012) consists of $39,209$ training and $12,630$ test images with 43 classes in total. We rescale the original images (see Figure 3) of varying size to 32×32 pixels and normalize them.

In order to perform a fair comparison between data augmentation and the use of a similarity loss, we make sure that every model is being shown the same amount of data in each training epoch. As a similarity loss model utilizes each input sample x twice in every training step under the transformations t_0 and t_1, we also present t_0 and t_1 to the data augmentation baseline in each training step. During training, data augmentation is performed online in a randomized manner. For rotated MNIST, t_0 and t_1 rotate the input x in every training step with an angle which is uniformly sampled between 0 and 2π radians.

In the case of GTSRB, we train for invariance to projective transformations, as traffic signs need to be correctly classified from different angles and distances. The augmentation with a projective transformation is performed by estimating an essential matrix using the eight-point algorithm from a set of point correspondences between the image corners and a randomized set of points. These points are randomly sampled from a uniform distribution within a distance of ±6 pixels in both dimensions of the image corners.

For all experiments we use the same randomization seeds for model comparisons but vary the seed across runs and for all experiments report the average numerical results over five independent runs.

4.2 Supervised Learning on Rotated MNIST Subset

For supervised learning, we integrate the similarity loss $\mathcal{L}_{sim}$ into an all convolutional network architecture (Springenberg et al., 2015) and use a subset of $N_s = 100$ labeled samples of the rotated MNIST dataset for training, where each class is represented equally often (*i.e.*, 10 times).

Our network closely resembles the CNN reference architecture for the rotated MNIST task in (Cohen and Welling, 2016). This network is constructed from seven convolutional layers, where each but the last layer uses filters of size 3×3 while the last layer uses filters of size 4×4. The convolutional filters are applied with a stride of 1×1. A max-pooling layer of stride and size 2×2 is inserted after the second convolutional layer. All but the last layer use batch normalization (Ioffe and Szegedy, 2015) before ReLU nonlinearities, followed by dropout with a keep probability of $p = 0.7$. On the last layer the softmax activation is applied. We use the Adam optimizer (Kingma and Ba, 2015) with a base learning rate of 0.001 and train with 100 samples per mini-batch.

In a first experiment, we evaluate the effect of applying the similarity loss on different layers l of the network (see Figure 4). We find that applying the similarity loss on the last layer results in the highest validation accuracy. This result is in line with findings by Cohen and Welling (2016), which showed that enforcing premature invariance in early layers of the network is undesirable. For all future experiments, we therefore only apply the similarity loss on the final output layer L of a network.

In addition, we perform a coarse hyperparameter search with a selected set of weight parameters $\lambda \in [1.0, 2.0, 3.0, 5.0, 7.5, 10.0, 15.0, 20.0]$. The results are plotted in Figure 5 and show improved validation accuracies for a wide range of λ values compared to using only data augmentation ($\lambda = 0.0$). While the performance is very robust to the choice of the weight parameter λ, we can observe a drop in validation accuracy when λ is large ($\lambda = 20.0$).

Table 1 confirms performance improvements on the test set for the proposed similarity loss with $\lambda = 5.0$ when training for 100 epochs with $N_s = 100$ labeled samples. Compared to a test error of 14.8% when training with data augmentation, we obtain an improved test error of 13.4% when training with an additional similarity loss. Additionally, we also obtain a better test error than our reimplementations of a harmonic network (Worrall et al., 2017) and a group-equivariant P4CNN (Cohen and Welling, 2016), which replace the regular convolutions in the network architecture with harmonic or group-equivariant convolutions, respectively.

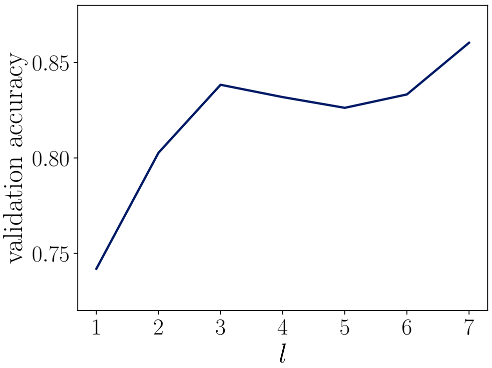


Figure 4: Hyperparameter study for the similarity loss layer l on the supervised rotated MNIST task.

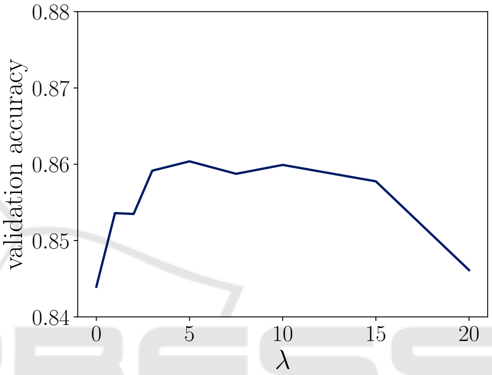


Figure 5: Hyperparameter study for the weight parameter λ on the supervised rotated MNIST task.

Table 1: Results for the supervised rotated MNIST task with $N_s = 100$.

Method	Test error (%)
Worrall et al. (2017)	21.5
Data augmentation	14.8
Cohen and Welling (2016)	14.2
Similarity loss	**13.4**

As a baseline we also evaluate the performance on the full dataset of 12,000 labeled examples. Here, no significant improvement is obtained by the similarity loss compared to a data augmentation model (see Table 2). Both data-driven methods are outperformed by harmonic networks (Worrall et al., 2017) and a group-equivariant P4CNN (Cohen and Welling, 2016).

Table 2: Results for the supervised rotated MNIST task with $N_s = 12,000$.

Method	Test error (%)
Data augmentation	3.7
Similarity loss	**3.6**
Cohen and Welling (2016)	2.28
Worrall et al. (2017)	1.69

Our results suggest that applying a similarity loss improves generalization and outperforms data augmentation as well as encoded transformation invariances when the number of labeled samples is small.

4.3 Semi-Supervised Learning

The unsupervised nature of the similarity loss $\mathcal{L}_{sim}$ makes it suitable as an additional guidance for semi-supervised learning problems in order to utilize unlabeled data during training.

4.3.1 Rotated MNIST

As a first architecture for semi-supervised learning on rotated MNIST, we use the convolutional architecture from Section 4.2 and a subset of $N_s = 100$ labeled samples. Additionally, we use the remaining training samples as unlabeled data. Each minibatch is constructed from 100 labeled and 100 unlabeled samples, where $\mathcal{L}_c$ is only applied on the labeled samples while $\mathcal{L}_{sim}$ is applied on the full minibatch. As before, we perform a hyperparameter study of the λ weight parameter from a set $\lambda \in [1.0, 2.0, 3.0, 5.0, 7.5, 10.0, 15.0, 20.0]$ (see Figure 6). When comparing to the λ-study for supervised learning (see Figure 5), we can now observe higher validation accuracies and again find the performance to be very robust.

Additionally, we perform a data ablation study where we vary the size of the labeled training set N_s. The results of the data ablation study are visualized in Figure 7. The figure demonstrates that the similarity loss is especially helpful when only very little labeled data is available. The benefit of a similarity loss (for a weight parameter of $\lambda = 10.0$ and 100 training epochs) is confirmed on the test set where the final test error is lowered by more than 2% compared to training only with data augmentation (see Table 3). Furthermore, the final test error is more than 1% lower compared to using the similarity loss on only the labeled images, which confirms the ability of the similarity loss to exploit additional unlabeled data.

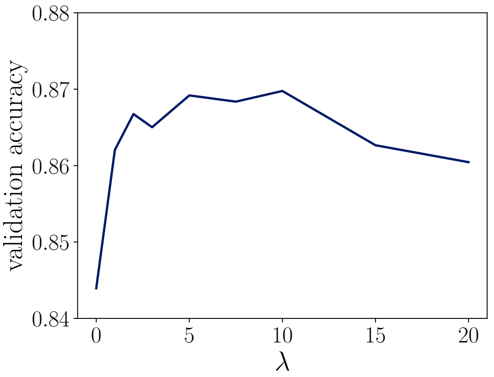


Figure 6: Hyperparameter study for λ on semi-supervised rotated MNIST.

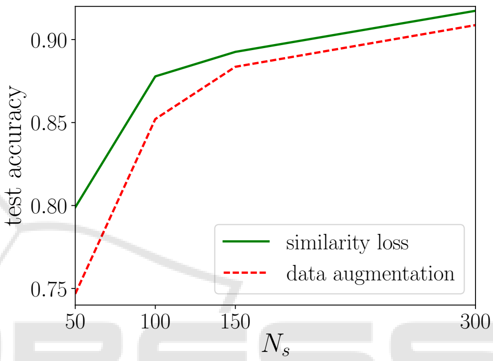


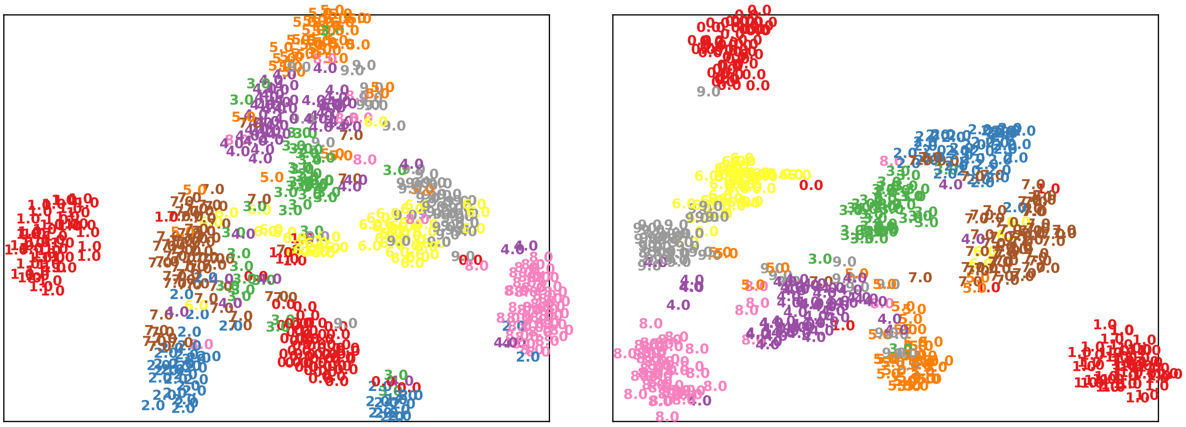
Figure 7: Accuracy vs. number of training samples on semi-supervised rotated MNIST.

Table 3: Results for the semi-supervised rotated MNIST task with $N_s = 100$.

Method	Test error (%)
Data augmentation	14.8
Similarity loss	**12.2**

We also observe improved class separability when visualizing the learned feature representations in the last layer of the model (see Figure 8).

For a second set of semi-supervised learning experiments, we incorporate the similarity loss into the fully connected ladder network architecture proposed by Rasmus et al. (2015) for the permutation invariant MNIST task. It features layers of size 784-1000-500-250-250-250-10 with respective denoising weight parameters $\alpha = [1000.0, 10.0, 0.10, 0.1, 0.1, 0.1, 0.1]$. The noisy encoder path uses Gaussian corruption noise with standard deviation 0.3. We train the network with mini-batches of 100 labeled and 256 unlabeled samples using the Adam optimizer and a base learning rate of 0.02 for 300 training epochs.



(a) Data Augmentation

(b) Similarity Loss

Figure 8: t-SNE visualizations of the learned feature representations on the final network layer.

Table 4: Ladder network results for semi-supervised rotated MNIST task with $N_s = 100$.

Method	Test error (%)
Data augmentation	8.0
Similarity loss (clean path)	**7.6**
Similarity loss (noisy path)	**6.8**

Table 4 displays the final test accuracies for utilizing our similarity loss in a ladder network with a weight parameter of $\lambda = 20.0$, which was determined in a separate hyperparameter search. We again find the addition of a similarity loss to be beneficial. Incorporating it in the noisy encoder path results in a better performance of 6.8% compared to a final test error of 7.6% in the clean encoder path. This can be explained by the Gaussian noise of the noisy encoder path providing additional regularization. Our results demonstrate that the use of a similarity loss also enables improving the performance of a previous state-of-the-art model architecture, specially designed for the semi-supervised learning task.

4.3.2 German Traffic Sign Recognition Benchmark

As a final experiment, we consider the German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2012). The network architecture for this task is an all-convolutional model, which resembles the All-CNN-C architecture proposed by Springenberg et al. (2015) for the CIFAR-10 task (Krizhevsky, 2009). It consists of nine convolution layers. The first four layers have 96, the, following layers 192 filters, all of size 3 × 3. They are applied with stride 1 except in the third and sixth layer where a stride of 2 is used. After the final convolutional layer average pooling is performed. Dropout is applied on the input with a probability of 0.2 and on the convolutional feature maps with 0.5. All layers use ReLU nonlinearities, batch normalization and weight decay of 0.001. A softmax activation is applied after the final layer. The network is trained with stochastic gradient descent and Momentum with mini-batches of 100 labeled and 100 unlabeled samples. A learning rate of 0.05 is decayed over the course of 100 training epochs.

In contrast with previous experiments on rotated MNIST, we now train for invariance to projective transformations. Unlike rotations, these cannot be easily encoded into the convolutional filters of a neural network. Here, the use of a similarity loss, which relies on augmenting the training data (as described in Section 4.1), offers a simple, yet effective solution to train CNNs for invariance to more complex geometric transformations.

Table 5: Results for the semi-supervised GTSRB task with $N_s = 2150$.

Method	Test error (%)
Data augmentation	14.8
Similarity loss	**9.6**

The test results for semi-supervised learning on the GTSRB task for 50 samples per class (*i.e.*, 2150 samples in total) and $\lambda = 10.0$ are displayed in Table 5. We observe a clear improvement in the final test accuracy from 14.8% to 9.6% when utilizing the similarity loss, despite the model already being heavily regularized by dropout, weight decay and batch normalization, which again indicates the effectiveness of the similarity loss when little labeled data is available.

As for rotated MNIST, we again perform a data ablation study (see Figure 9). The study shows that the improvement when using an additional similarity loss is largest when N_s is small, but that even for lar-

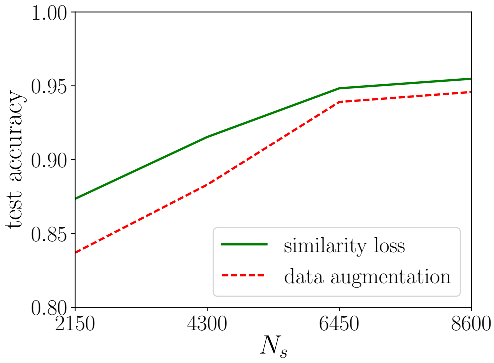


Figure 9: Accuracy vs. number of training samples on semi-supervised GTSRB.

ger sizes of the labeled training set the similarity loss outperforms the data augmentation model.

5 DISCUSSION

The experiments in Section 4 demonstrate the benefits of using the proposed similarity loss in both supervised and semi-supervised learning tasks. Data augmentation works well in practice for fully supervised problems when big labeled training sets are available. However, in this work, we show that an additional similarity loss can act as an effective regularizer, which improves upon data augmentation when little annotated training data is available. The benefit of the proposed similarity loss is not limited to the "toy-like" rotated MNIST task but extends to more complex geometric transformations of natural images where even larger improvements can be obtained.

In addition, another contribution of our work concerns the fact that the proposed similarity loss can be utilized for semi-supervised learning, where it can help to exploit additional unlabeled data. The inclusion of the similarity loss in a semi-supervised ladder network shows particular promise. With our proposed modification, we further improve over an architecture which until recently was the state-of-the-art approach for semi-supervised learning. As the rotated MNIST dataset has not been commonly used to evaluate semi-supervised learning architectures we do not claim to set a new state-of-the-art but consider the ladder network using an additional similarity loss to have highly competitive performance.

In general, the use of an unsupervised similarity loss is a surprisingly simple idea which can easily be integrated into any deep learning model. All it requires is to duplicate the classification stream of the network and tune the λ hyperparameter. In our experiments, we found the performance to generally be very robust to the choice of the weight parameter λ. The similarity loss can be additionally combined with methods which encode invariances directly into convolutional filters (Cohen and Welling, 2016; Worrall et al., 2017; Zhou et al., 2017) and with architectures which enable neural networks to handle geometric transformations more easily (Jaderberg et al., 2015; Dai et al., 2017).

6 CONCLUSIONS

This work proposes an unsupervised similarity loss which penalizes differences between the predictions for transformed copies of an input for improved learning of transformation invariance in deep neural networks on the rotated MNIST and German Traffic Sign Recognition Benchmark classification tasks. We show that our similarity loss acts as an effective regularizer, which improves model performance when little annotated data is available, in both supervised and semi-supervised learning. Future work could investigate the application of the proposed similarity loss on a combination of network layers or an adjustment of the weight parameter λ over the course of training.

While this work improves the use of data-driven methods based on augmenting training data for learning transformation invariance, there still remains a gap to techniques which encode invariances to transformations directly into the filters of a convolutional neural network when training on the full set of labels on rotated MNIST (Cohen and Welling, 2016; Worrall et al., 2017; Zhou et al., 2017). However, unlike the proposed similarity loss, which can easily be applied for a wide variety of transformations, these methods are currently limited to simple geometric transformations such as rotations or mirror reflections.

A promising avenue for future research is therefore the development of approaches which encode invariances to more complex geometric transformations directly into the architecture of deep neural networks and combine them with soft constraints as presented in this paper in order to further improve the data efficiency of deep neural networks.

REFERENCES

Agrawal, P., Carreira, J., and Malik, J. (2015). Learning to see by moving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

Bruna, J. and Mallat, S. (2013). Invariant scattering convolution networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 35(8):1872–1886.

Cohen, T. S. and Welling, M. (2016). Group equivariant convolutional networks. In *Proc. of the International Conf. on Machine learning (ICML)*.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. *Arxiv tech report*.

Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*.

Haeusser, P., Mordvintsev, A., and Cremers, D. (2017). Learning by association - a versatile semi-supervised training method for neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of the International Conf. on Machine learning (ICML)*.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*.

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

Kivinen, J. J. and Williams, C. K. I. (2011). Transformation equivariant boltzmann machines.

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto.

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

Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. (2016). TI-POOLING: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proc. of the International Conf. on Machine learning (ICML)*.

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

Miyato, T., Maeda, S., Koyama, M., Nakae, K., and Ishii, S. (2016). Distributional smoothing by virtual adversarial examples. In *Proc. of the International Conf. on Learning Representations (ICLR)*.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. of the European Conf. on Computer Vision (ECCV)*.

Oyallon, E. and Mallat, S. (2015). Deep roto-translation scattering for object classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. (2015). Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*.

Sifre, L. and Mallat, S. (2013). Rotation, scaling and deformation invariant scattering for texture discrimination. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis.

Sohn, K. and Lee, H. (2012). Learning invariant representations with local transformations. In *Proc. of the International Conf. on Machine learning (ICML)*.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. A. (2015). Striving for simplicity: The all convolutional net. In *International Conf. on Learning Representations (ICLR) (workshop track)*.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Zheng, S., Song, Y., Leung, T., and Goodfellow, I. (2016). Improving the robustness of deep neural networks via stability training. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, Y., Ye, Q., Qiu, Q., and Jiao, J. (2017). Oriented response networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.