POST-HOC DOMAIN ADAPTATION VIA GUIDED DATA HOMOGENIZATION

Kurt Willis & Luis Oala
Fraunhofer HHI
Berlin, Germany
{kurt.willis,luis.oala}@hhi.fraunhofer.de

ABSTRACT

Addressing shifts in data distributions is an important prerequisite for the deployment of deep learning models to real-world settings. A general approach to this problem involves the adjustment of models to a new domain through transfer learning. However, in many cases, this is not applicable in a post-hoc manner to deployed models and further parameter adjustments jeopardize safety certifications that were established beforehand. In such a context, we propose to deal with changes in the data distribution via guided data homogenization which shifts the burden of adaptation from the model to the data. This approach makes use of information about the training data contained implicitly in the deep learning model to learn a domain transfer function. This allows for a targeted deployment of models to unknown scenarios without changing the model itself. We demonstrate the potential of data homogenization through experiments on the CIFAR-10 and MNIST data sets.

A routine assumption in machine learning is that data in the training and testing environment is identically distributed. In reality, data distributions may differ (Hendrycks et al., 2020), leading to performance degradations for models trained with standard deep learning algorithms. Transfer learning (Pan & Yang, 2010) has emerged as a field that aims to transfer knowledge of models from a learned source task to a target task by exploiting underlying commonalities. Domain adaptation in particular, addresses a mismatch in either the input space $\mathcal{X}$ or the data distribution $\mathbb{P}$ of a source and target task: $(\mathcal{X}_s, \mathbb{P}_s) \neq (\mathcal{X}_t, \mathbb{P}_t)$. A common approach to this challenge is to encourage the model to learn a domain invariant representation of the data. This is typically done through the use of the joint distribution, requiring access to the source data set.

This work proposes domain adaptation through homogenization of the data in a post-hoc-manner. This eliminates the need to anticipate numerous possible test-time scenarios during training and the original model is kept intact - a desirable property for security and robustness certification (Balunovic et al., 2019; Oala et al., 2020). In detail, an explicit domain mapping function is learned through an optimization objective adopted from work by Yin et al. (2020). This does not assume access to the source data set; instead, the data statistics measured in an appropriate feature space are sufficient. These statistics are readily available in networks that make use of the widely adapted batch normalization layers (Ioffe & Szegedy, 2015).

An example for the relevance of post-hoc domain adaptation is seen in optical systems which provide inputs for deep learning applications. Sensor corrosion or re-calibration of hardware devices has proven to be a significant barrier to the reliable application of deep learning systems in fields such as medicine (Heaven, 2020) or autonomous driving (Michaelis et al., 2020).

We show the ability to homogenize non-identically distributed data in experiments on the CIFAR-10 (Krizhevsky, 2012) and MNIST (LeCun & Cortes, 2010) data sets. The method is further tested in the unsupervised setting, where no labeled data in the target domain is available.

Related work focuses on learning domain-invariant representations in order to bridge the discrepancy in distributions. In this effort, Ghifary et al. (2015) propose training an auto-encoder that is able to encode both domains. Tzeng et al. (2015) make use of the criterion loss along with a domain confusion loss which is implemented by adding an adaptation layer on top of the classifier's last layer. This auxiliary layer aims to make the learned representations indistinguishable between domains by

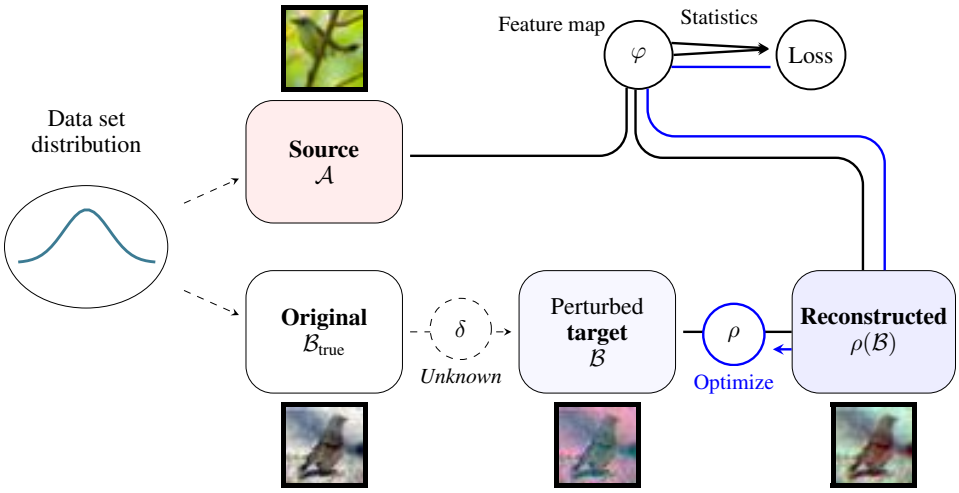


Figure 1: Overview of the homogenization setting. A general transformation ρ is optimized, so that a neural network Φ regains its performance on the transformed dataset $\rho(\mathcal{B})$. The feature maps are obtained from Φ.

maximally confusing a domain classifier. Long et al. (2015) further extend the use of adaptation layers to multiple layers throughout the network. Zellinger et al. (2019) and Sun & Saenko (2016) formulate a distribution loss incorporating higher-order moments. Saito et al. (2019) propose to classify data of the target domain by comparing feature representations to prototypes for each class. Zhu et al. (2017) propose CycleGAN, a generative adversarial network that learns explicit domain transformations, by formulating an adversarial loss and measuring reconstruction accuracy.

This work can also be viewed in the context of inverse problems, and more specifically, related to the task of deblurring or deconvolution. Whereas Xu et al. (2014) and Kobler et al. (2017) train a deconvolution model for image reconstruction on handcrafted perturbation functions, this work solely makes use of the prior knowledge learned by an image classification network and statistics to correct for general distribution shifts that also entail deconvolutions.

1 METHOD

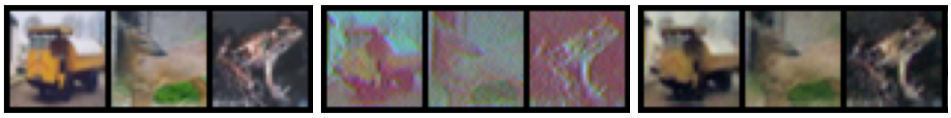
Figure 1 summarizes the main idea of guided data homogenization. Suppose we are given a neural network that was trained on - or performs reasonably well on - a source data set $\mathcal{A}$ and a target data set $\mathcal{B}$, typically smaller in size ($n_{\mathcal{B}} < n_{\mathcal{A}}$). Further, the distribution of $\mathcal{B}$ differs crucially from $\mathcal{A}$, i.e. the neural network is unable to attain high performance on data coming from $\mathcal{B}$. The task is to learn a transformation ρ that adjusts $\mathcal{B}$ so that the network's performance is regained.

An underlying assumption is that $\mathcal{B}$ originates from the same distribution as $\mathcal{A}$, but has been corrupted or transformed by an unknown perturbation δ. The transformation ρ is modeled to correct for this perturbation, in the sense that $\rho(\mathcal{B}) \approx \mathcal{B}_{\text{true}}$. This is done by minimizing a dissimilarity-score between $\rho(\mathcal{B})$ and $\mathcal{A}$ in the feature space given by an appropriate feature map φ. Towards this goal, simple statistics (mean and variance) are recorded for both data sets after they have been transformed by φ. By use of backpropagation, ρ is optimized, in order to make the statistics of the target $\mathcal{B}$ approach those of the source $\mathcal{A}$. Whether matching statistics in feature space actually increases similarity between $\mathcal{A}$ and $\mathcal{B}$ strongly depends on φ and its representation of the data set.

For a given feature mapping $\varphi : \mathbb{R}^d \to \mathbb{R}^m$, the dissimilarity score, or the **statistics-loss**, is defined as follows.

$$L_\varphi(\mathcal{A}, \mathcal{B}) = \| \mu(\varphi(\mathcal{A})) - \mu(\varphi(\mathcal{B})) \|_2 + \| \sigma^2(\varphi(\mathcal{A})) - \sigma^2(\varphi(\mathcal{B})) \|_2 \tag{1}$$

μ and σ^2 denote the empirical feature mean and variance over the feature channels, as is generally done in the computations of BatchNorm-layers. A data set mapping is obtained by applying a feature map φ to every input: $\varphi(\mathcal{A}) := \{(\varphi(\mathbf{x}), y) \mid (\mathbf{x}, y) \in \mathcal{A}\}$. This notion can be extended to a



(a) Data set $\mathcal{A}$ (b) Data set $\mathcal{B}$ (c) Homogenized $\mathcal{B}$

Figure 2: Excerpt of results on the homogenization experiments on CIFAR10 data set in a supervised setting after 100 epochs for NN ALL method. $\kappa = 0.15$, $n_{\mathcal{B}} = 512$. The full results are depicted in Figure 3 of the Appendix.

collection of feature maps: $L_{[\varphi_1,\ldots,\varphi_n]}(\mathcal{A},\mathcal{B}) = \sum_{i=1}^{n} L_{\varphi_i}(\mathcal{A},\mathcal{B})$. Furthermore, a **class-dependent** loss can be obtained by splitting the data set into subsets for each class and accumulating the resulting losses: $L_{\varphi}^{CC}(\mathcal{A},\mathcal{B}) = \sum_{c=1}^{C} L_{\varphi}(\mathcal{A}|_c, \mathcal{B}|_c)$. Here, $\mathcal{A}|_c = \{(\mathbf{x}, y) \in \mathcal{A} \mid y = c\}$ is the subset of $\mathcal{A}$ constrained to samples of label c. This case can be compared to related work, where class-prototypes are used.

2 Experiments

The experiments[1] are conducted on the popular image recognition data sets MNIST and CIFAR-10. The target data set $\mathcal{B}$ is obtained by applying a randomized, parameter-controlled perturbation to a small unseen subset of the original data set. The **perturbation model** is a composition of additive Gaussian noise and two noise-controlled 2d-convolutions: $\delta(\mathbf{x}) = \mathbf{K}_{\kappa}^{(2)}(\mathbf{K}_{\kappa}^{(1)}(\mathbf{x} + \kappa\boldsymbol{\mu})$. The kernel matrices of the convolutions are each of size 3×3 and initialized to the identity kernel with added Gaussian noise. The overall noise level is controlled by the parameter κ. The **reconstruction model** is a vanilla four-layer convolutional residual network of width 16.

Feature maps

As explained in Section 1, the statistics-loss depends on the feature mapping φ. Various feature mappings are compared in the experiments. The main approach (**NN ALL**) involves all hidden states of a neural network $\Phi = (\Phi_{\mathrm{L}} \circ \ldots \Phi_1)$. These are seen as the outputs of a collection of feature maps $[\varphi_0, \ldots, \varphi_{\text{L-1}}]$, where

$$\begin{aligned}\varphi_\ell &: \mathbb{R}^d \to \mathbb{R}^{d_\ell} = (\Phi_\ell \circ \ldots \Phi_1)\,, \text{ for } \ell = 1, \ldots, L-1\\ \varphi_0 &: \mathbb{R}^d \to \mathbb{R}^d \;\; = \mathrm{id}\,.\end{aligned}$$

Another feature mapping serving as a comparison is induced by the penultimate layer $\Phi_{\mathrm{L}-1}$, the layer before the logits layer. It is reported as **NN**.

$$\varphi : \mathbb{R}^d \to \mathbb{R}^{d_{\text{L-1}}} = (\Phi_{\text{L-1}} \circ \cdots \circ \Phi_1)$$

In order to further delineate the benefit of using representations stemming from the neural network, a feature mapping involving only one linear layer with a width of 512 is considered. Comparing the performance of the reconstruction using shallow versus deep features can further warrant the necessity of complex feature maps. This one linear layer can be seen as a set of random linear projections; it is reported as **RP**.

The hidden representations of the same neural network model with randomly initialized parameters (**RANDOM NN**) serves as another baseline comparison. This is done in order to test the utility of an optimized feature representation.

The complete loss formulation for the supervised setting is then given by

$$r_{\text{stats}} L_{\varphi}(\mathcal{A},\mathcal{B}) + r_{\text{crit}} L_{\text{crit}}(\mathcal{B}, y_{\mathcal{B}})\,,$$

where $y_{\mathcal{B}}$ are the target labels and L_{crit} is the criterion-loss, which, in this case, is the cross-entropy loss. The unsupervised setting only makes use of L_{φ} (i.e. $r_{\text{crit}} = 0$).

[1]The code for guided data homogenization and all experiments can be accessed at `https://github.com/willisk/Thesis`.

Table 1: Metrics of homogenization results in the supervised setting after 100 optimization epochs for CIFAR-10. The mean and the standard deviation over 5 runs are given. The number of samples $n_{\mathcal{B}} = 512$.

Data set	**accuracy** [%]	**validation accuracy** [%]	**verification accuracy** [%]	↓ **l2-error**	↑ **PSNR**	↑ **SSIM** [%]
Data set $\mathcal{A}$	99.7	–	99.8	0	∞	100
Data set $\mathcal{B}_{\text{true}}$	99.8	93.2	99.8	0	∞	100
Data set $\mathcal{B}$	26.2	26.6	26.0	24.3	16.5	84.0
Homogenized $\mathcal{B}$						
CRITERION	**99.6 ± 0.4**	77.9 ± 2.1	75.9 ± 2.7	14.0 ± 2.7	18.0 ± 0.4	86.7 ± 2.2
NN	95.5 ± 0.6	83.1 ± 0.6	80.8 ± 0.8	11.6 ± 1.6	19.5 ± 0.3	90.1 ± 0.5
NN ALL	99.3 ± 0.4	**88.2 ± 0.6**	**84.9 ± 1.0**	**10.0 ± 2.6**	**23.3 ± 0.3**	**95.4 ± 0.2**
RANDOM NN	51.9 ± 5.0	48.4 ± 5.7	42.7 ± 4.2	29.0 ± 10.6	11.3 ± 0.7	59.6 ± 4.7
RP	51.1 ± 2.1	51.0 ± 1.4	47.3 ± 1.1	76.4 ± 5.2	18.1 ± 0.2	82.6 ± 0.6

3 RESULTS

For measuring the overall distortion of the reconstruction, the **l2-error** ε_2 of the unit vectors is calculated. $\varepsilon_2^2 = \frac{1}{d}\sum_{i=0}^{d} \|\rho(\delta(\mathbf{e}_i)) - \mathbf{e}_i\|_2^2$, where $\mathbf{e}_i$ is the i-th unit vector of the standard basis. Further, a metric commonly encountered in image quality assessment, the **peak signal-to-noise ratio** (**PSNR**) is reported. One last metric, that doesn't rely on the mean-squared error, the **structural similarity index measure** (**SSIM**) (Horé & Ziou, 2010) is considered. It is re-scaled to fit the interval $[0, 1]$ and is given as a percentage (%).

In order to measure the generalization ability of the learned transformation, a second, independently trained neural network's (Φ_{ver}) accuracy is reported as the **verification accuracy**. Further, the same transformations are applied to an unseen validation set $\mathcal{C}$. The accuracy attained by Φ_{ver} on this new set constitutes the **validation accuracy**.

The baseline evaluation metrics, along with the final results of the reconstruction task on the CIFAR-10 data set are given in Table 1 for each method. Less than 1% of the original training set sample size is used in these experiments. After the perturbation has been applied to $\mathcal{B}$ the scores drop significantly. Relying solely on the criterion results in overfitting the reconstruction function, as is seen by the disparity in generalization scores. Random projections (RP) and the randomly initialized convolutional neural network fail to capture vital information about the data set distribution; although they are still able to improve various scores. The best-performing method is NN ALL.

While shallow features are able to perform the reconstruction task on simple data sets like MNIST (see Table 4 of the Appendix), their benefit on a more complex data set like CIFAR-10 is limited. Further results for the class-conditional formulation L^{CC} are given in Appendix A. Using this formulation, a significant performance boost is noted when $r_{\text{crit}} = 0$, however, no notable benefit is obtained when the criterion loss is included.

The evaluation metrics for the unsupervised setting, where no label information is provided are given in Table 3 of the Appendix. The scores are close to the supervised case and show promising results even for small sample sizes.

4 CONCLUSIONS

Neural networks, and convolutional neural networks in particular for image data sets, are powerful feature extractors. The learned feature representations carry implicit knowledge about the training data, and by tracking simple statistics of the latent states, more information about the data distribution is gained. Yin et al. (2020) have shown that it is possible to recover high-fidelity, plausible images from only having access to the BatchNorm-statistics. This work demonstrates that it is also possible to use this information in the context of domain adaptation for data homogenization and, in particular, shows its applicability to deconvolution or re-calibration tasks. Guided data homogenization proves to be viable even when few data in the target domain is available or even when no label information in the target domain is given. Further exploration of the connection between implicit model knowledge and post-hoc model adaptation offers an intriguing avenue for future research.

REFERENCES

Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 15313–15323. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f7fa6aca028e7ff4ef62d75ed025fe76-Paper.pdf.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Will Douglas Heaven. Google's medical ai was super accurate in a lab. real life was a different story. — mit technology review, April 2020. URL https://technologyreview.com.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2020.

Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. pp. 2366–2369, 08 2010. doi: 10.1109/ICPR.2010.579.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Erich Kobler, Teresa Klatzer, Kerstin Hammernik, and Thomas Pock. Variational networks: Connecting variational methods and deep learning. In Volker Roth and Thomas Vetter (eds.), *Pattern Recognition*, pp. 281–293, Cham, 2017. Springer International Publishing. ISBN 978-3-319-66709-6.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Yann LeCun and Corinna Cortes. Mnist handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming, 2020.

Luis Oala, Jana Fehr, Luca Gilli, Pradeep Balachandran, Alixandro Werneck Leite, Saul Calderon-Ramirez, Danny Xie Li, Gabriel Nobis, Erick Alejandro Muñoz Alvarado, Giovanna Jaramillo-Gutierrez, Christian Matek, Arun Shroff, Ferath Kherif, Bruno Sanguinetti, and Thomas Wiegand. Ml4h auditing: From paper to practice. In *Proceedings of the Machine Learning for Health NeurIPS Workshop*, volume 136 of *Proceedings of Machine Learning Research*, pp. 280–317. PMLR, 11 Dec 2020. URL http://proceedings.mlr.press/v136/oala20a.html.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8050–8058, 2019.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4068–4076, 2015.

Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 1790–1798. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/1c1d4df596d01da60385f0bb17a4a9e0-Paper.pdf.

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning, 2019.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A FULL RESULTS

A.1 CIFAR-10

Table 2: Metrics on homogenization results in the supervised setting after 100 optimization epochs for CIFAR-10, including the class-conditional formulation (CC). The mean and the standard deviation over 5 runs are given. $n_{\mathcal{B}} = 512$.

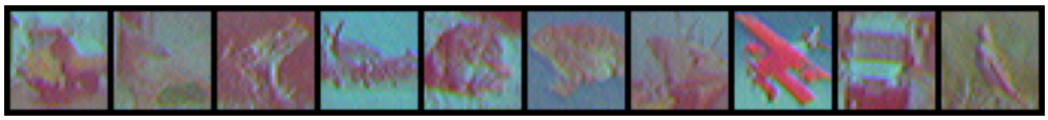
Homogenized $\mathcal{B}$	**accuracy** [%]	**validation accuracy** [%]	**verification accuracy** [%]	↓ **l2-error**	↑ **PSNR**	↑ **SSIM** [%]
CRITERION	99.6 ± 0.4	77.9 ± 2.1	75.9 ± 2.7	14.0 ± 2.7	18.0 ± 0.4	86.7 ± 2.2
NN	95.5 ± 0.6	83.1 ± 0.6	80.8 ± 0.8	11.6 ± 1.6	19.5 ± 0.3	90.1 ± 0.5
NN CC	$\mathbf{99.9 \pm 0.1}$	85.8 ± 0.7	83.0 ± 0.6	$\mathbf{9.9 \pm 2.2}$	20.9 ± 0.6	92.0 ± 0.9
NN ALL	99.3 ± 0.4	$\mathbf{88.2 \pm 0.6}$	$\mathbf{84.9 \pm 1.0}$	10.0 ± 2.6	23.3 ± 0.3	$\mathbf{95.4 \pm 0.2}$
NN ALL CC	99.8 ± 0.0	87.8 ± 0.9	84.7 ± 1.0	12.8 ± 0.9	$\mathbf{23.7 \pm 0.5}$	$\mathbf{95.4 \pm 0.4}$
RANDOM NN	51.9 ± 5.0	48.4 ± 5.7	42.7 ± 4.2	29.0 ± 10.6	11.3 ± 0.7	59.6 ± 4.7
RANDOM NN CC	55.6 ± 1.6	53.0 ± 2.7	48.1 ± 1.9	38.2 ± 10.8	13.5 ± 0.6	67.4 ± 4.3
RP	51.1 ± 2.1	51.0 ± 1.4	47.3 ± 1.1	76.4 ± 5.2	18.1 ± 0.2	82.6 ± 0.6
RP CC	42.6 ± 0.9	42.5 ± 0.9	41.4 ± 1.2	42.2 ± 1.7	17.0 ± 0.2	81.5 ± 0.4

Table 3: Metrics on homogenization results in the unsupervised setting after 100 optimization epochs for CIFAR-10. The mean and the standard deviation over 5 runs are given. $n_{\mathcal{B}} = 512$.

Homogenized $\mathcal{B}$	**accuracy** [%]	**validation accuracy** [%]	**verification accuracy** [%]	↓ **l2-error**	↑ **PSNR**	↑ **SSIM** [%]
NN	85.6 ± 1.3	79.8 ± 1.7	76.4 ± 0.3	18.1 ± 2.3	17.4 ± 0.7	85.3 ± 1.0
NN ALL	$\mathbf{91.8 \pm 0.6}$	$\mathbf{84.5 \pm 0.7}$	$\mathbf{83.6 \pm 1.0}$	$\mathbf{14.1 \pm 2.3}$	$\mathbf{22.1 \pm 0.6}$	$\mathbf{94.5 \pm 0.3}$
RANDOM NN	57.1 ± 6.4	53.2 ± 4.9	49.3 ± 5.4	34.3 ± 17.8	12.1 ± 0.7	67.2 ± 3.4
RP	47.7 ± 1.1	45.8 ± 1.0	42.8 ± 0.9	36.1 ± 5.1	15.5 ± 0.3	79.0 ± 0.7

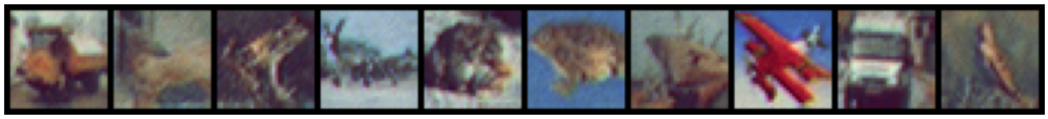
Ground Truth

Distorted



CRITERION

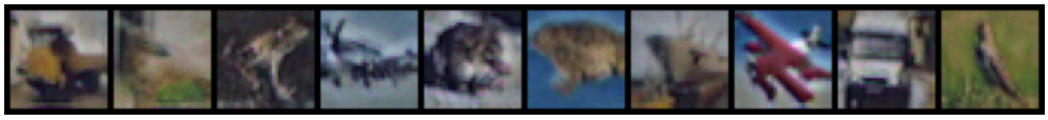
NN



NN CC

NN ALL

NN ALL CC



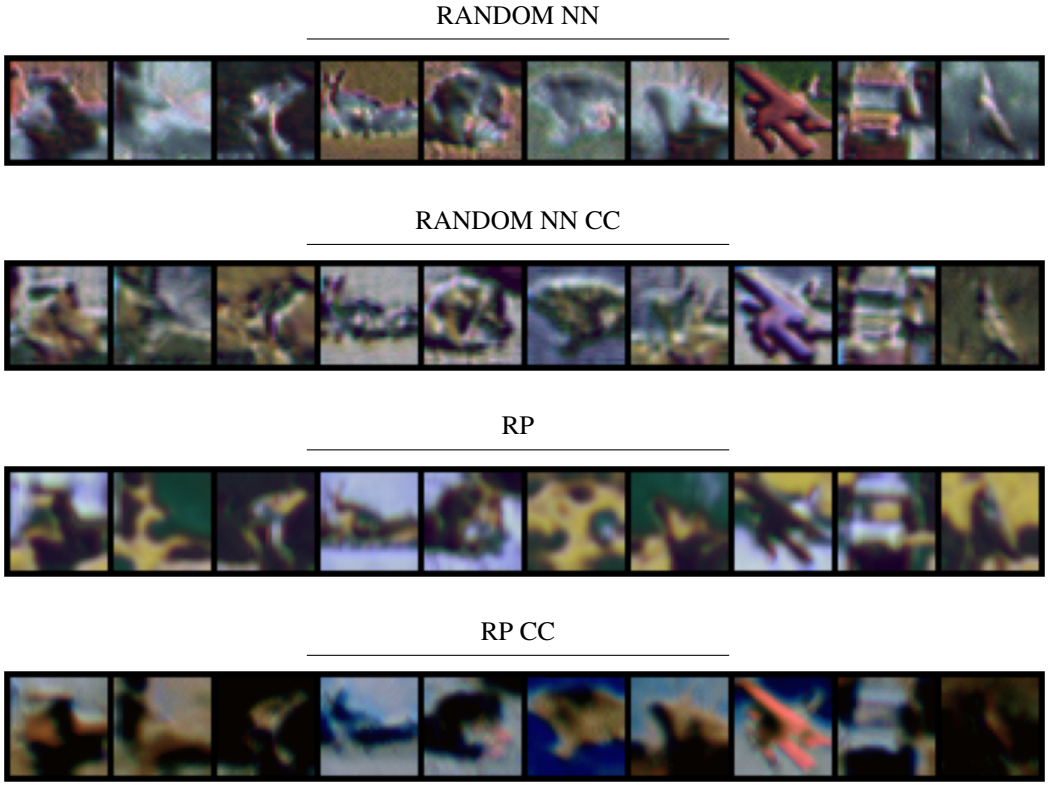


Figure 3: Results of the homogenization on CIFAR-10 data set after 100 epochs

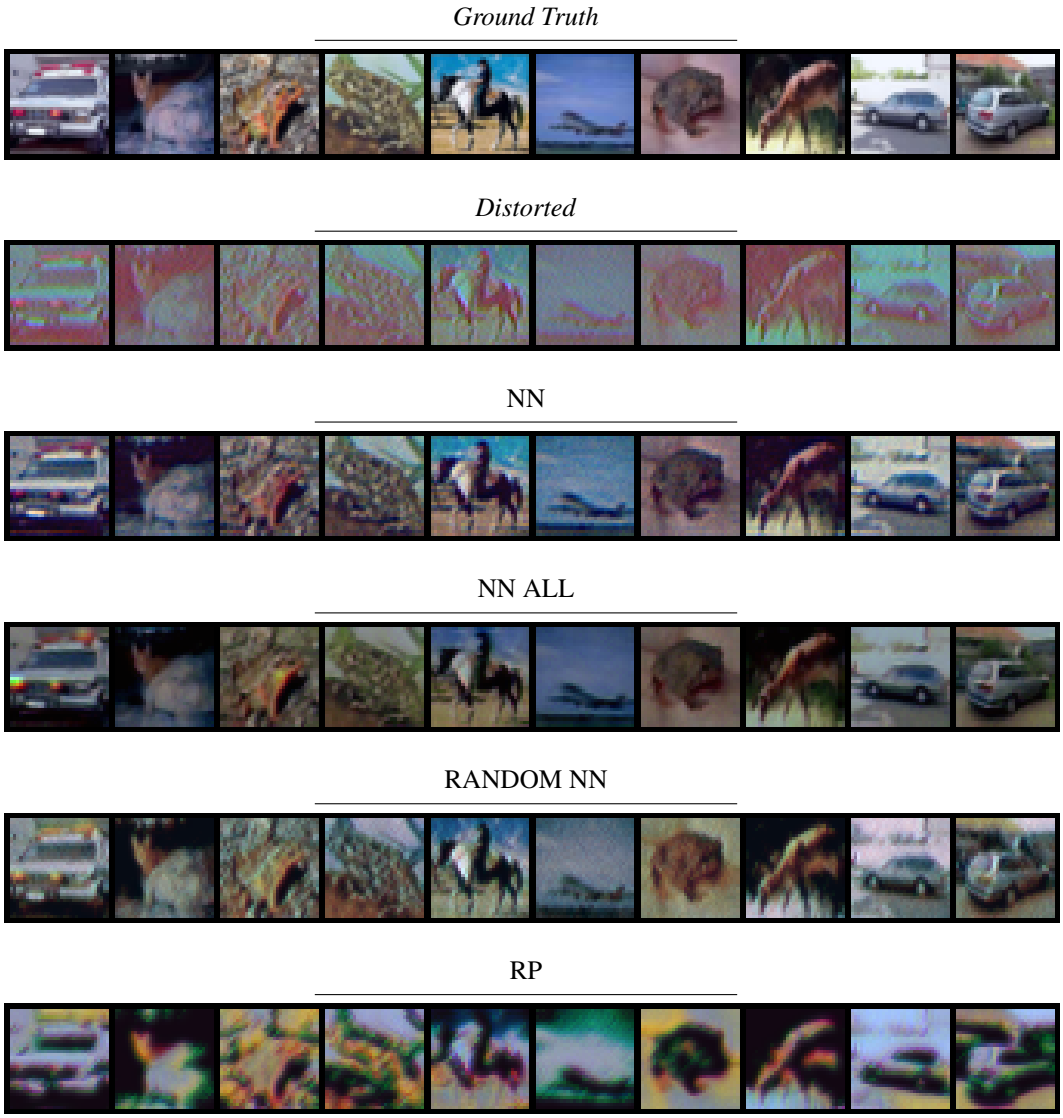


Figure 4: Results of the homogenization on CIFAR-10 data set in an unsupervised setting after 100 epochs.

A.2 MNIST

Table 4: Metrics on homogenization results after 100 optimization epochs for MNIST. The mean and the standard deviation over 5 runs are given. $n_{\mathcal{B}} = 512$.

Homogenized $\mathcal{B}$	**accuracy** [%]	**validation accuracy** [%]	**verification accuracy** [%]	↓ **l2-error**	↑ **PSNR**	↑ **SSIM** [%]
CRITERION	**99.8 ± 0.0**	97.7 ± 0.0	90.4 ± 0.2	9.8 ± 0.1	15.7 ± 0.0	80.6 ± 0.0
NN	97.5 ± 0.2	98.0 ± 0.1	89.7 ± 1.9	16.9 ± 0.5	18.0 ± 0.1	88.5 ± 0.4
NN CC	97.9 ± 0.2	98.2 ± 0.1	97.8 ± 0.3	14.3 ± 1.1	20.1 ± 0.4	91.9 ± 1.6
NN ALL	98.3 ± 0.3	98.1 ± 0.3	**98.0 ± 0.9**	9.5 ± 1.7	21.2 ± 0.1	90.0 ± 1.1
NN ALL CC	97.7 ± 0.3	**98.3 ± 0.2**	96.7 ± 0.4	13.2 ± 1.1	**22.2 ± 0.1**	**93.3 ± 1.0**
RANDOM NN	**99.8 ± 0.0**	97.7 ± 0.1	90.3 ± 0.7	9.9 ± 0.3	15.8 ± 0.0	80.7 ± 0.0
RANDOM NN CC	**99.8 ± 0.0**	97.8 ± 0.1	90.5 ± 0.4	10.0 ± 0.2	15.8 ± 0.0	80.7 ± 0.0
RP	**99.8 ± 0.0**	98.0 ± 0.1	97.3 ± 0.1	**7.2 ± 0.1**	15.3 ± 0.0	81.0 ± 0.0
RP CC	**99.8 ± 0.0**	97.9 ± 0.0	96.6 ± 0.1	7.6 ± 0.2	15.4 ± 0.0	81.0 ± 0.0

Ground Truth

Distorted

CRITERION

NN

NN CC

NN ALL

NN ALL CC

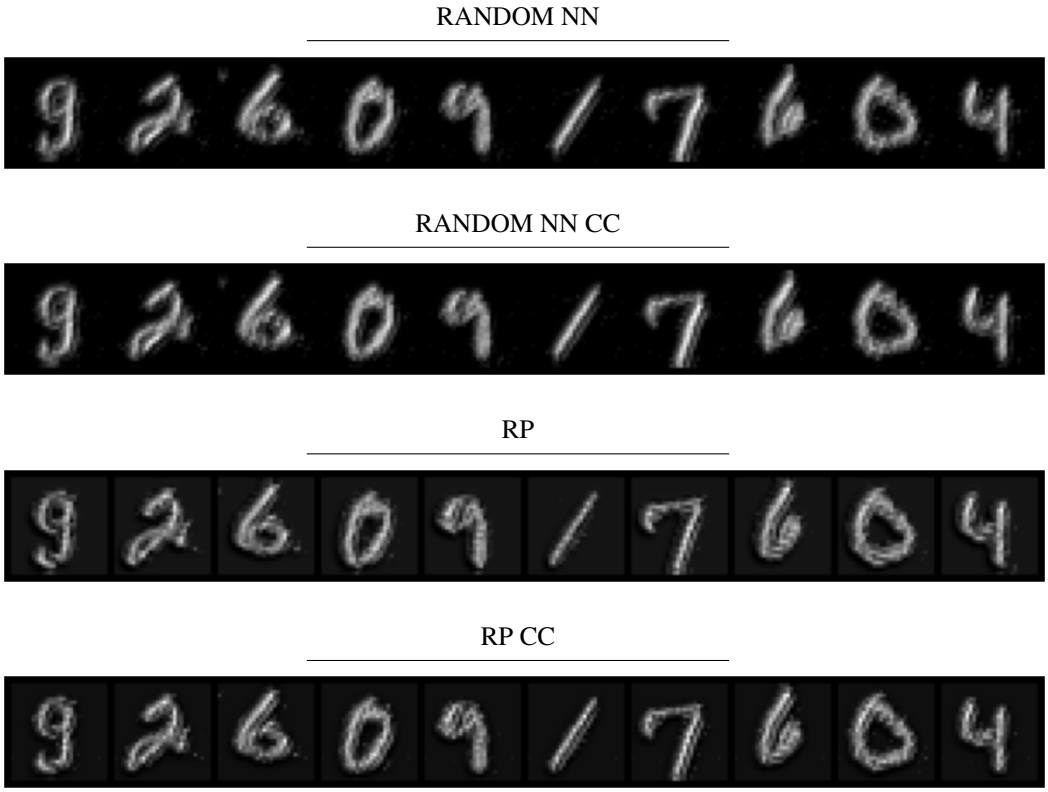


Figure 5: Results of the reconstruction task on MNIST data set after 100 epochs

B PARAMETER-SETTINGS

Table 5: Parameters and specifications of the experiments. $n_{\mathcal{A}}$, $n_{\mathcal{B}}$, $n_{\mathcal{C}}$ denote the sample size of the source, target and validation data set respectively. C is the number of classes and d the number of input dimensions of the data set. κ is the distortion parameter of the perturbation model. s_{width} and s_{width} are the width and the depth of the residual network used for reconstruction. s_{RP} is the number of random projections, the width used for RP. r_{crit} and r_{stats} are the loss weighting factors of the objective function.

MNIST	
Data Set	
$n_{\mathcal{A}}$	= 60,000
$n_{\mathcal{B}}$	= 512
$n_{\mathcal{C}}$	= 1024
Properties	
C	= 10
input shape	= (1, 28, 28)
d	= 784
Distortion	
κ	= 0.3
Neural Network Architecture	
main network:	ResNet20
verifier network:	ResNet9
Reconstruction Model	
s_{width}	= 8
s_{depth}	= 8
s_{RP}	= 512
Optimization	
learning rate	= 0.1
batch size	= 128
r_{crit}	= 1
r_{stats}	= 0.001

CIFAR10	
Data Set	
$n_{\mathcal{A}}$	= 50,000
$n_{\mathcal{B}}$	= 512
$n_{\mathcal{C}}$	= 1024
Properties	
C	= 10
input shape	= (3, 32, 32)
d	= 3072
Distortion	
κ	= 0.15
Neural Network Architecture	
main network:	ResNet34
verifier network:	ResNet18
Reconstruction Model	
s_{width}	= 8
s_{depth}	= 8
s_{RP}	= 512
Optimization	
learning rate	= 0.1
batch size	= 128
r_{crit}	= 1
r_{stats}	= 10