Invariance-inducing regularization using worse-case transformations suffices to boost accuracy and spatial robustness

Fanny Yang [1 2] **Zuowen Wang** **Christina Heinze-Deml** [3]

Abstract

This work provides theoretical and empirical evidence that adding an invariance-inducing regularizer to standard and adversarial training increases predictive accuracy for worst-case spatial transformations (spatial robustness). In fact, with the same computational budget, it achieves a relative reduction of the spatially robust error of 20% for CIFAR10, even surpassing hand-crafted spatial-equivariant networks. For SVHN, we additionally observe that regularized training improves both the standard test and robust accuracy against spatial transformations.

1. Introduction

As deployment of machine learning (ML) systems in the real world has steadily increased over recent years, more and more emphasis is placed on the reliability of the algorithms. It is for example important to understand the properties of commonly used neural networks including invariances to certain types of perturbations from both a security and interpretability point of view.

As expressivity of neural networks has been shown both theoretically [15, 3, 12] and empirically [42],in this paper we study how we can find good predictors that are also invariant against small spatial transformations like rotations and translations. We measure invariance by evaluating the *robust loss*, that is prediction performance on worst-case transformations (*attacks*) of each test image. This robust (also called *adversarial*) loss has recently gathered a lot of attention in the field of adversarially robust learning [34, 25, 19, 29, 5, 22, 32, 33, 31, 38, 24], motivated by security considerations.

As a basis for our proposed regularized methods, we observe in Appendix B that when perturbations are from transformation groups, predictors that optimize the robust loss are in fact invariant. Although recent works show a fundamental trade-off between robust and standard accuracy in constructed ℓ_∞ perturbation settings (see e.g. [36, 43]), we prove that this is fundamentally different for spatial transformations due to their group structure.

For deep neural networks, there are two main approaches to achieve invariance and robustness (used interchangeably in this paper): one relies on regularization using additionally artificially modified training data [2, 41], such as adversarial training or plain data augmentation. A more transformation specific approach is to carefully design specialized architectures to incorporate spatial equivariance [16, 20, 7, 23, 39, 37, 11, 35].

Given the above two lines of work, we want to explore the following questions: (a) how far can we push augmentation based methods for common deep neural networks to be more robust against spatial transformations? (b) how robust are specialized architectures designed to achieve invariance against entire transformation groups?

In order to explore (a) and (b) we implemented various training methods with an explicit invariance-inducing regularizer as described in Section 2 and compare spatial robustness for these augmentation-based methods and specialized neural network architectures and SVHN test images. We find that regularized methods can achieve $\sim 20\%$ relative adversarial error reduction compared to previously proposed augmentation-based methods (including adversarial training) without requiring additional computational resources. They empirically even outperform a few traditional spatial-equivariant networks on CIFAR10 and SVHN. Finally, we observe that on SVHN, not only does the robust test accuracy increase with invariance-promoting regularization but it helps to boost standard accuracy as well.

2. Experimental setup

We now present the details of our methods that we compare in terms of robustness against spatial transformations.

[1]Department of Statistics, Stanford [2]Institue of Theoretical Studies, ETH Zurich [3]Department of Mathematics, ETH Zurich. Correspondence to: F. Yang, C. Heinze-Deml <fannyang, heinzedeml@stat.math.ethz.ch>.

ICML 2019 Workshop on Understanding and Improving Generalization in Deep Learning, Long Beach, California, 2019. Copyright 2019 by the author(s).

2.1. Regularization based methods

Denoting by $\mathcal{B}(x)$ the perturbation (also called *transformation invariant*) set of images that result from translations of maximally 3 pixels and rotations of at most 30 degrees, we consider SGD applied to the following loss functions

$$\min_{f\in\mathcal{F}} \mathcal{L}_{nat}(f;R,\lambda) := \mathbb{E}\,\ell(f(X),Y) + \lambda R(f,X,Y)$$
$$\min_{f\in\mathcal{F}} \mathcal{L}_{rob}(f;R,\lambda) := \mathbb{E}\sup_{x'\in G^X} \ell(f(x'),Y) + \lambda R(f,X,Y).$$

where the expectation is with respect to the empirical distribution and we consider the following different choices for R that are encouraging invariance

- $R_{\ell_2}(f,X,Y) = \sup_{x'\in\mathcal{B}(x)} \|f(X) - f(x')\|_2^2$
- $R_{\text{KL}}(f,X,Y) = \sup_{x'\in\mathcal{B}(x)} \mathrm{D}_{\text{KL}}(f(x'),f(X))$ (same as [43] for $\mathcal{L}_{\text{nat}}$)
- $R_{\text{ALP}}(f,X,Y) = \|f(x') - f(X)\|_2^2$ with $x' = \arg\max_{u\in\mathbb{G}^X} \ell(f(u),Y)$ (ALP[18])

We describe the different methods we used to calculate the supremum over the perturbation set in Section C.1 under *defenses* and refer to the maximizers of both regularizers and cross entropy loss as *adversarial examples*.

2.2. Attacks and defenses

The attacks and defenses we choose largely follow the setup in [10]. Recall that the defense refers to the (efficient) procedure at training time which aims to make the resulting model robust to adversarial examples and generally differs from the (extensive) attack mechanism performed at evaluation time to assess the model's robustness.

Grid search attack Following [10], during test time we use grid search with 775 distinct combinations of translation and rotations to find the most adversarial example in the perturbation set for each test image. Although projected gradient descent methods are effective in increasing adversarial test accuracy for ℓ_∞ perturbations, we confirm previous observations [10] that they are worse than grid search in this small dimensional perturbation set.

Defenses For the adversarial example which maximizes either the loss or regularization function, we use the following defense mechanisms:

- Wo-k: the worst of k randomly drawn transformation
- S-PGD: Several steps of Projected Gradient Descent with respect to the spatial transformation parameters.
- Random: A randomly drawn transformation

More details about the attack and defense methods can be found in Section C.1 in the Appendix.

For clarity of presentation of our different methods we use a naming convention consisting of the following components: (a) `Reg` : refers to what regularizer was used (ALP, ℓ_2, KLor AT) where ATrefers to standard unregularized adversarial training(b) `batch`: refers to using the gradient of the loss on the adversarial examples (rob), natural examples (nat) or both (mix), and (c) `def`: the mechanism used to find the adversarial example, including random (rnd), worst-of-10 (Wo-10) and spatial PGD (S-PGD) as described in Section C.1. Thus, `Reg` (`batch`, `def`) corresponds to using `Reg` as a regularizing function, the examples defined by `batch` in the gradient of the loss and the defense `def` to find the inner maximizer.

2.3. Spatial equivariant networks

We compare the robust prediction accuracies from the regularized networks with three specialized architectures G-ResNet, ETN and STN, designed to be equivariant against spatial transformations and translations. (see Section C.2 in the Appendix for more details).

3. Empirical Results

This section is devoted to comparing the natural test accuracy (standard accuracy on the test set) and test grid accuracy (as defined in Section C.1) achieved by standard, regularized (adversarial) training techniques and specialized spatial equivariant architectures described in Section 2.3.

In Table 1 we report results for a subset of the experiments to facilitate comparisons. Results for a more complete set of methods can be found in Appendix, Tables 7–11. For each method, we report averages (standard errors are contained in Tables 7–11) computed over five training runs with identical hyperparameter settings. We compare all methods by computing absolute and relative error reductions (defined as $\frac{\text{absolute error drop}}{\text{prior error}}$). It is insightful to present both numbers since the absolute values vary drastically between datasets.

The experiments are conducted choosing deep neural networks as the function space $\mathcal{F}$ and ℓ is the cross-entropy loss. In the main paper we consider the datasets SVHN [28] and CIFAR-10. For the regularized methods, we train a ResNet-32 [13], implemented in TensorFlow [1]. The precise training procedure is described in Sec. C.3. For training with adversarial examples we augment the data using random lef-tright flips and add random translations of ± 4px for non-adversarially trained models. .

Effectiveness of regularization for adversarial training In Table 1 (top), the three leftmost columns represent unregularized methods which all perform worse in grid accuracy than regularized methods. When considering the three regu-

Table 1. Mean accuracies of models trained with various forms of regularized adversarial training as well as standard augmentation techniques (top) and spatial equivariant networks (bottom). $\star$ denotes standard augmentation plus random rotations.

	std	std*	AT (rob, Wo-10)	KL (rob, Wo-10)	ℓ_2 (rob, Wo-10)	ALP (rob, Wo-10)	KL-C (mix, S-PGD)	ALP (rob, S-PGD)
SVHN (nat)	95.48	93.97	96.03	96.13	96.53	96.30	96.14	96.11
(grid)	18.85	82.60	90.35	92.71	92.55	92.04	92.42	92.32
CIFAR (nat)	92.11	89.93	91.76	90.41	90.53	90.11	89.98	89.85
(grid)	9.52	58.29	71.17	77.47	77.06	75.9	78.93	77.80

	GRN	GRN*	ETN	ETN*	STN	STN*
SVHN (nat)	96.07	95.05	95.53	95.57	95.61	95.55
(grid)	25.12	84.9	13.15	84.21	36.68	79.28
CIFAR (nat)	93.39	93.08	–	–	–	–
(grid)	16.85	71.64	–	–	–	–

larizers with the same `batch` and `def` (here chosen to be "rob" and Wo-10) regularized adversarial training improves the grid accuracy from 71.17% to 77.47% on CIFAR-10 and 90.35% to 92.71% on SVHN, corresponding to a relative error reduction of 22% and 24% respectively.

Computational considerations In Figure 1, we plot the grid accuracy vs. the runtime (in hours) for a subset of regularizers and defense mechanisms on CIFAR-10 for clarity of presentation. How much overhead is needed to obtain the reported gains? Comparing AT(rob, Wo-k) and ALP(rob, Wo-k) shows that significant improvements in grid accuracy can be achieved by regularization with only a small computational overhead. What if we make the defense stronger? Improving the adversarial example search in unregularized methods by increasing k to 20 is not effective: AT(rob, Wo-20) achieves 72.31% instead of 71.17% grid accuracy for AT(rob, Wo-10) while requiring $3\times$ more training time. Although regularizing with KL generally achieves the best results, the adversarial example search requires two forward passes and is hence slow. Hence, we consider KL-C(mix, S-PGD) which computes the adversarial example with respect to the classification cross entropy loss like ALP. This approach achieves a similar accuracy as KL(nat, S-PGD) while being $2\times$ faster. In fact, this variant achieves the best overall grid accuracy of 78.93% at a similar runtime as AT(rob, Wo-20) with 72.93%, translating into a relative error reduction of 22%.

Comparison with spatial equivariant networks Although the rotation-augmented G-ResNet44 obtains higher grid (84.9%, 71.64%) and natural accuracies ($95\%, 93.08\%$) than the rotation-augmented Resnet-32 on both SVHN ($82.60\%/93.97\%$) and CIFAR10 ($58.29\%/89.93\%$), regularizing augmentation (i.e. regularizers with "rnd", see Table 2 (right)) using both the ℓ_2 distance and the KL divergence matches the G-ResNet44 on CIFAR10 ($71.60\%, 73.50\%$ respectively) and surpasses it

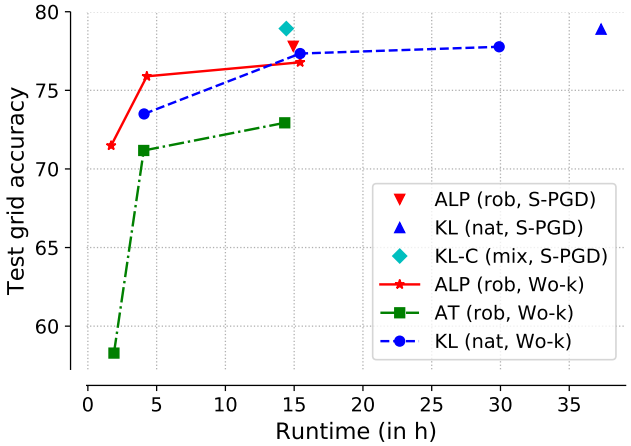

Figure 1. Mean runtime for different methods on CIFAR-10. The connected points correspond to Wo-k defenses with $k \in \{1, 10, 20\}$.

on SVHN on grid ($90.51\%, 90.69\%$) and natural accuracies by a relative grid error reduction of $\sim 37\%$. The same phenomenon is observed for the augmented ETN and STN on SVHN. In conclusion, invariance-oblivious regularized augmentation based methods match or outperform state-of-the-art end-to-end networks handcrafted to be equivariant to spatial transformations. relative error

Trade-off natural vs. adversarial accuracy SVHN is one of the main datasets (without artificial augmentation like in rot-MNIST [21]) where spatial equivariant networks have reported improvements on natural accuracy. This is due to the inherent orientation variance in the data. In our mathematical framework, this corresponds to the assumption of the marginal conditional distribution of an transformations of an image having support on the transformation-invariant set. Furthermore, as all numbers in SVHN have the same label irrespective of small rotations of at most 30 degrees, the second assumption in Theorem 2 is also fulfilled. Table 1

Table 2. Mean accuracies of models trained with various forms of regularized adversarial training. Left: All adversarial examples were found via Wo-10; right: data augmentation setting.

	KL (nat, Wo-10)	ℓ_2 (nat, Wo-10)	ALP (nat, Wo-10)	ℓ_2 (nat, rnd)	KL (nat, rnd)	ℓ_2 (rob, rnd)	KL (rob, rnd)
SVHN (nat)	96.00	96.05	96.39	96.34	96.16	96.09	96.23
(grid)	92.27	92.16	91.98	90.51	90.69	90.48	90.92
CIFAR (nat)	90.83	88.32	88.78	87.80	89.33	88.75	89.47
(grid)	77.34	75.64	75.43	71.60	73.50	71.49	73.22

confirms the statement in the Theorem that improving robust accuracy may not hurt natural accuracy or even improve it. A similar trend can be seen for CIFAR10 for which the methods with highest grid accuracies such as KL(rob, Wo-10) also have close to the highest natural accuracy.

Comparing the different regularizers and $\mathcal{L}_{\text{nat}}(f; R)$ vs. $\mathcal{L}_{\text{rob}}(f; R)$ We are interested in disentangling the underlying reasons behind differences between the grid accuracy performances of different combinations of regularization methods. Tables 1 and 2 shed some light on this quest. In particular we can observe that, fixing any regularizer, the robust regularized loss `Reg` (rob, `def`) (i.e., $\mathcal{L}_{\text{rob}}(f; R)$) does better (or not statistically significantly worse) than `Reg` (nat, `def`) (i.e., $\mathcal{L}_{\text{nat}}(f; R)$). Second, the KL regularizer is better than ℓ_2 for a large number of settings.

Choice of λ The different regularization methods peak at different λ. However, they outperform unregularized methods in a large range of λ values, suggesting that well-performing values of λ are not difficult to find in practice. These can be seen in Figures 4 and 3 in the Appendix

Many more insightful experiments can be found in Section C.6.

4. Related work

Group equivariant networks There are in general two types of approaches to incorporate spatial invariance into the network. In one of the earlier works in the neural net era, Spatial Transformer Networks were introduced [16] which includes of a transformer module that predicts transformation parameters followed by a transformer. Later on, one line of work proposed multiple filters that are discrete group transformations of each other [20, 23, 7, 45, 39]. For continuous transformations, steerability [37, 8] and coordinate transformation [11, 35] based approaches have been suggested.

Regularized training Using penalty regularization to encourage robustness and invariance when training neural networks has been studied in different contexts: for distributional robustness [14], domain generalization [26], ℓ_p adversarial training [27, 18, 43], robustness against simple transformations [6] and semi-supervised learning [44, 40]. These approaches are based on augmenting the training data either *statically* [14, 26, 6, 40], ie. before fitting the model, or *adaptively* in the sense of adversarial training, with different augmented examples per training image generated in every iteration [18, 27, 43].

Robustness against simple transformations Approaches targeting adversarial accuracy for simple transformations have used attacks and defenses in the spirit of PGD (either on transformation space [10] or on input space projecting to transformation manifold [17]) and simple random or grid search [10, 30]. Recent work [9] has also evaluated some rotation-equivariant networks with different training and attack settings which reduces direct comparability with e.g. adversarial based defenses [10].

5. Conclusion

In this work, we have explored how regularized augmented and adversarial training compares against specialized spatial equivariant networks in terms of robustness against spatial transformations consisting of translations and rotations. Strikingly, even though the method of augmentation can be applied to encourage *any* desired invariance, the regularized methods adapt well and perform similarly or better than specialized networks. We showed that for transformation invariances and under certain practical assumptions on the distribution, there is no trade-off between natural and adversarial accuracy which stands in contrast to the debate around ℓ_p-perturbation sets. In summary, it is advantageous to replace unregularized with regularized augmented/adversarial training in a large variety of setings.

6. Acknowledgements

We thank Nicolai Meinshausen and Ludwig Schmidt for helpful discussions and Luzius Brogli for experimental contributions. FY was supported by the Institute for Theoretical Studies ETH Zurich, the Dr. Max Rössler and Walter Haefner Foundation and the Office of Naval Research Young Investigator Award N00014-19-1-2288.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Henry S Baird. Document image defect models. In *Structured Document Image Analysis*, pages 546–556. Springer, 1992.

[3] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[4] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[6] Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2019.

[7] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pages 2990–2999, 2016.

[8] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *International Conference on Learning Representations*, 2018.

[9] Beranger Dumont, Simona Maggio, and Pablo Montalvo. Robustness of rotation-equivariant networks to adversarial perturbations. *arXiv preprint arXiv:1802.06627*, 2018.

[10] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, 2019.

[11] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. In *International Conference on Learning Representations*, 2018.

[12] Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *arXiv preprint arXiv:1708.02691*, 2017.

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

[14] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *arXiv preprint arXiv:1710.11469*, 2017.

[15] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial Transformer Networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[17] Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

[18] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[19] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[20] Dmitry Laptev, Nikolay Savinov, Joachim M Buhmann, and Marc Pollefeys. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 289–297, 2016.

[21] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 473–480. ACM, 2007.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[23] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *ICCV*, pages 5058–5067, 2017.

[24] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3575–3583, 2018.

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[26] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 3, 2017.

[27] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. In *International Conference on Learning Representations*, 2018.

[28] Y. Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 5, 2011.

[29] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[30] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*, 2017.

[31] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

[32] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

[33] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[34] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[35] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant Transformer Networks. *arXiv preprint arXiv:1901.11399*, 2019.

[36] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[37] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[38] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018.

[39] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[40] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation, 2019.

[41] Larry S. Yaeger, Richard F. Lyon, and Brandyn J. Webb. Effective training of a neural network character classifier for word recognition. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 807–816. MIT Press, 1997.

[42] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2015.

[43] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *International Conference of Machine Learning*, 2019.

[44] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4488, 2016.

[45] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 519–528, 2017.

A. Supplementary material

B. Theoretical results

In this section, we first introduce our notion of transformation sets to formalize robustness against a small range of translations and rotations. We then prove that on a population level constraining or regularizing for transformation invariance yields models that minimize the robust loss. Moreover, under reasonable assumptions on the true distribution, we show the set of robust minimizers not only minimizes the natural loss, but under mild conditions on the distribution over the transformations, is even equivalent to the set of natural minimizers.

Although the framework can be applied to general problems and transformation groups, we consider image classification for concreteness. In the following, $X \in \mathbb{R}^d$ are the observed images and $Y \in \mathbb{R}^p$ the one-hot vector for multiclass labels and both are random variables from a joint distribution $\mathbb{P}$. The function $f : \mathbb{R}^d \to \mathbb{R}^p$ in function space $\mathcal{F}$ (e.g. deep neural network in experiments) maps the input image to a logit $x \to f(x)$ that is then used for prediction via a softmax layer.

B.1. Transformation sets

Invariance against spatial transformations is often thought of in terms of group equivariance of the representation and prediction. Instead of invariance with respect to all spatial transformations in a group, we impose a weaker requirement, that is invariance against transformation sets, defined as follows. We denote by G^z a compact subset of images in the support of $\mathbb{P}$ that can be obtained by transformation of an image $z \in \mathcal{X}$ in the space of all possible observable images $\mathcal{X}$. G^z is called a *transformation set* and by requiring the technical assumption on the space of real images that the sampling operator is bijective, the mapping $z \to G^z$ is also bijective. We can hence define $\mathcal{G}$, a set of transformation sets, by $\mathcal{G} = \cup_{z \in \mathcal{X}} G^z$ for a given transformation group. It is important to note that G^z are disjoint for different images $z \in \mathcal{X}$. More details on the aforementioned concepts and definitions can be found in Sec. B.5.

The above definition is distribution dependent and $\mathcal{G}$ partitions the support of the distribution $\widetilde{\mathcal{X}}$. As a consequence, each $x \in \widetilde{\mathcal{X}}$ is uniquely mapped to an element in $\mathcal{G}$ which we denote by $G^x \in \mathcal{G}$. We say that a function f is *(transformation-)invariant* if $f(x) = f(x')$ for all $x, x' \in U$ for all $U \in \mathcal{G}$. Equivalently, the set of all invariant functions can be described by

$$\mathcal{V}(R) := \{f : R(f, x) = 0 \quad \forall x \in \widetilde{\mathcal{X}}\}$$

where $R(f, x) = \sup_{x' \in G^x} h(f(x), f'(x))$ for any semimetric h on $\mathbb{R}^p$.[1] For all choices of semimetrics the spaces $\mathcal{V}(R)$ are identical which we denote by $\mathcal{V}$. Using this notation, fitting a model with high accuracy under "small" transformations of the input can be mathematically captured by the robust optimization formulation [4] of minimizing the *robust loss*

$$\mathcal{L}_{\text{rob}}(f) := \min_{f \in \mathcal{F}} \mathbb{E}_{X,Y} \sup_{x' \in G^X} \ell(f(x'), Y). \tag{1}$$

We call the solution of this problem the *robust* minimizer. While adversarial training aims to optimize the empirical version of Eq. (1), the converged predictor might be far from the global population minimum in particular in the case of nonconvex landscapes for neural networks. Furthermore, we show in the following section that for robustness against transformation sets, constraining the model class to invariant functions leads to the same optimizer of the robust loss. These facts motivate invariance-inducing regularization in the next section that we then show to exhibit improved robust test accuracy in practice.

B.2. Regularization to encourage invariance

In the sequel, $\mathcal{F}$ is the function space of the models that we are using to fit the data. Instead of the specific definition of R via semimetrics we now allow for general R and consider constrained optimization problems of the form

$$\min_{f \in \mathcal{F}} \mathbb{E}\, \ell(f(X), Y) \text{ s.t. } f \in \mathcal{V}(R) \tag{O1}$$

$$\min_{f \in \mathcal{F}} \mathbb{E} \sup_{x' \in G^X} \ell(f(x'), Y) \text{ s.t. } f \in \mathcal{V}(R). \tag{O2}$$

The following theorem shows that the optimization programs (O2) are equivalent to solving (1) if the set of all invariant functions $\mathcal{V}$ is a subset of the function space $\mathcal{F}$.

[1] The weaker notion of a semimetric satisfies almost all conditions for a metric without having satisfying the triangle inequality.

Theorem 1. *If $\mathcal{V} \subset \mathcal{F}$, all minimizers of the adversarial loss* (1) *are in $\mathcal{V}$. If furthermore $\mathcal{V}(R) \subseteq \mathcal{V}$, any solution of the optimization programs* (O1)*,* (O2) *minimizes the adversarial loss.*

The proof of Theorem 1 can be found in the Appendix in Sec. B.6. Since projection onto the constrained set is not computationally feasible for neural network function spaces, a common workaround is to relax the constraints by only requiring $f \in \{f : R(f, x) \leq \epsilon \quad \forall x \in \widetilde{\mathcal{X}}\}$. Then, by Lagrangian duality, (O1) and (O2) can be rewritten in penalized form as

$$\begin{aligned} \min_{f \in \mathcal{F}} \mathcal{L}_{nat}(f; R, \lambda) &:= \min_{f \in \mathcal{F}} \mathbb{E}\, \ell(f(X), Y) + \lambda R(f, X) \\ \min_{f \in \mathcal{F}} \mathcal{L}_{rob}(f; R, \lambda) &:= \min_{f \in \mathcal{F}} \mathbb{E} \sup_{x' \in G^X} \ell(f(x'), Y) + \lambda R(f, X) \end{aligned} \tag{2}$$

for some scalar $\lambda > 0$. In Sec. B.4 we discuss how ordinary adversarial training and modified variants that have been proposed thereafter can be viewed as special cases of (2).

B.3. Trade-off between natural and robust accuracy

Even though low robust accuracy (1) might be the main goal in some applications, one might wonder whether the robust minimizer exhibits lower accuracy on untransformed images (*natural accuracy*) defined as $\mathcal{L}_{\text{nat}}(f) := \mathbb{E}_{\,X,Y} \ell(f(X), Y)$ [36, 43]. In this section we tackle this question and identify the conditions for transformation based perturbations under which minimizing robustness does not lead to decreased natural accuracy. Under mild assumptions it even increases.

One reason why adversarial examples have attracted a lot of interest is because the prediction of a given classifier can change in a perturbation set in which all images appear the same to the human eye. Mathematically, in the case of transformation sets, the latter can be modeled by a property of the true distribution, namely that the conditional distribution Y given x is constant for all x belonging to the same subset $U \in \widetilde{\mathcal{G}}$, or in other words, Y is conditionally independent of X given G^X, i.e. $Y \perp\!\!\!\perp X | G^X$. Under this assumption one can show that there is no trade-off in natural accuracy for the transformation robust minimizer.

Theorem 2 (Trade-off natural vs. robust accuracy)**.** *Under the assumption of Theorem 1 and if $Y \perp\!\!\!\perp X | G^X$ holds, the adversarial minimizer also minimizes the natural loss. If moreover, the probability distribution of X conditioned on every transformation set G^z has support G^z and the loss ℓ is injective, then every minimizer of the natural loss also has to be invariant.*

As a consequence, minimizing the constrained optimization problem (O1) could potentially help in finding the optimal solution to the unconstrained loss minimization. Practically, the assumption on the distribution on the transformation sets G^z corresponds to inherent transformation variance in the natural distribution of the dataset. This could be a possible explanation why we observe a boost in natural accuracy for robust invariance-inducing methods in Sec. 3 on SVHN, a commonly used benchmark dataset for spatial-equivariant networks for this reason.

One might wonder how this result relates to several recent publications such as [43], [36] that presented toy examples for which the ℓ_∞ robust solution must have higher natural loss than the Bayes optimal solution. On a fundamental level, ℓ_∞ perturbation sets are of different nature compared to transformation sets on generic distributions of $\mathcal{X}$. In the distribution they consider, there is no unique mapping from $x \in \mathcal{X}$ to a perturbation set and thus the conditional independence property cannot generally hold.

B.4. Different regularizers and practical implementation

In order to improve robustness against spatial transformations we consider different choices of $R(f, x)$ in the regularized objective (2) that we then compare in Sec. 3. First of all, we relax the definition to include explicit dependence on y which allows us to view common adversarial training as a special variant of our regularized framework. In particular, common adversarial training as in [34, 22] corresponds to $\mathcal{L}_{nat}$ with $R(f, X, Y) = \sup_{x' \in G^X} \ell(f(x'), Y) - \ell(f(X), Y)$. In this paper we compare common adversarial training with regularized variants that use the following regularizers

- $R_{\ell_2}(f, X, Y) = \sup_{x' \in G^X} \|f(X) - f(x')\|_2^2$
- $R_{\text{KL}}(f, X, Y) = \sup_{x' \in G^X} \mathrm{D}_{\text{KL}}(f(x'), f(X))$ (equivalent to [43] for $\mathcal{L}_{\text{nat}}$)[2]

[2]Notice that [43] is actually using the KL divergence in their implementation, which is not the same as cross-entropy as claimed in the

- $R_{\text{KL-C}}(f, X, Y) = \mathrm{D}_{\text{KL}}(f(x'), f(X))$ with $x' = \arg\max_{u \in \mathbb{G}^X} \ell(f(u), Y)$
- $R_{\text{ALP}}(f, X, Y) = \|f(x') - f(X)\|_2^2$ with $x' = \arg\max_{u \in \mathbb{G}^X} \ell(f(u), Y)$ (see [18])

where D_{KL} is the KL divergence on the softmax of the vectors $f \in \mathbb{R}^p$. Furthermore, standard data augmentation can also be used in conjunction with a regularizer by adding $R(f, X) = \mathbb{E}_{\ x' \in G^X} h(f, X, X')$ for any semimetric h.

All of the above regularizers involve a maximization over the invariance set. In the first two cases the maximization is performed with respect to the *regularization function* h while in the latter two regularizers the maximum is taken with respect to the loss function ℓ. In all cases we refer to the maximizer as an *adversarial example*. For the regularizers $R_{\ell_2}, R_{\text{KL}}$ the assumption in Theorem 1 is satisfied and hence the solution of the corresponding constrained problem (O1) is a minimizer of the adversarial loss. In contrast, for ALP and KL-C, the statements of the theorems do not hold since $\mathcal{V}(R) \not\subset \mathcal{V}$.

Adversarial example for spatial transformation sets Since G^X is not a closed group and we do not know whether the drawn $X \in G^X$ lies at the boundary of G^X or in the interior, we cannot solve the maximization constrained to G^X. However, for an appropriate choice of set $\mathcal{S}$ (see Sec. B.5 in the Appendix), we can instead minimize an upper bound of (1) which reads

$$\min_{f \in \mathcal{F}} \mathbb{E} \sup_{\Delta \in \mathcal{S}} \ell(f(\mathcal{T}(X, \Delta)), Y) \geq \min_{f \in \mathcal{F}} \mathbb{E} \sup_{x' \in G^X} \ell(f(x'), Y) \tag{3}$$

where $\mathcal{S}$ is the set of transformations that we search over and $\mathcal{T}(X, \Delta)$ denotes the transformed image with transformation Δ. The left hand side in (3) is hence what we aim to solve in practice where the expectation is over the empirical joint distribution of X, Y. The relaxation of G^X to a range of transformations of X that is $\{\mathcal{T}(X, \Delta) \ : \Delta \in \mathcal{S}\}$ is also used for the maximization within the regularizers.

B.5. Rigorous definition of transformation sets and choice of $\mathcal{S}$

Let us define a "potential" image as a function $I : \mathbb{R}^2 \to [0, 255]^3$ in the function space $\mathcal{I}$. For practical purposes the images become finite-dimensional vectorized matrices by sampling functions in $\mathcal{I}$ and hence some of the properties that allow clean handling of spatial transformations only hold somewhat approximately.

We define transformation sets as subsets of the function space $\mathcal{I}$. Let us define the symmetric group $\mathbb{G}$ of all rotations and horizontal and vertical translation acting on $\mathcal{I}$. It consists of elements g_Δ uniquely parameterized by $\Delta \in \mathbb{R}^3$ and can be represented by a coordinate transform matrix, see e.g. [7]. The first two dimensions represent the values for the translations and the third represents rotation.

The transformed image (function) $g_\Delta(I) \in \mathcal{I}$ can be expressed by $g_\Delta(I)(v) = I(G^{-1}v)$ for each $v \in \mathbb{R}^2$ where G_Δ is the coordinate transform matrix associated with $\Delta \in \mathbb{R}^3$ as in [7]. For each $I \in \mathcal{I}$, the group orbit is $\mathbb{G}^I = \{g(I) : g \in \mathbb{G}\}$. By definition of $\mathcal{I}$, the group orbits partition the space $\mathcal{I}$.

Subsets of orbits For our problem setting, requiring invariance in the entire orbit (i.e. with respect to all translations and rotations) is too restrictive. First of all, large transformations rarely occur in nature because of physical laws and common human perspectives (an upside down tower for example). Secondly, in image classification robustness is usually only required against adversarial attacks which would not fool humans, i.e. lead them to mislabel the image. If the transformation set is too large, this requirement is not fulfilled anymore.

For this purpose we now define compact subsets of $\mathbb{G}^I$

$$\mathbb{G}_{\mathcal{S}}(I) = \{g_\Delta(I) : \Delta \in \mathcal{S}_I \subset \mathbb{R}^3\} \tag{4}$$

where $\mathcal{S}_I$ could be different for every $I \in \mathcal{I}$. By assumption on $\mathcal{I}$, the subsets $\mathbb{G}_{\mathcal{S}}(I)$ are disjoint. In fact, they partition $\mathcal{I}$.

We now define for each I the set of "maximal" transformations $\Delta_I^\star$ such that $(-\Delta_I^\star, \Delta_I^\star) \supset \mathcal{S}_I$. Now define $\Delta^\star = \arg\max_{\Delta_I^\star} \|\Delta_I^\star\|$ to be the largest transformation over all images.

Note that the conventional definitions for invariance or equivariance in the group-sense as in [7] do not carry over to subsets of orbits. Our definition of an invariant function $f : \mathcal{X} \to \mathbb{R}^d$ is instead

$$f(I') = f(I) \qquad \forall I', I \in \mathbb{G}_{\mathcal{S}}(I)$$

paper.

For the group of rotation and translation on images, this property translates to f having the same value for all images that are only slightly translated and/or rotated versions of each other.

Sampling issues In reality, the observed image is not a function on $\mathbb{R}^2$ but a vector $z \in \mathbb{R}^{w\times h}$ that is the result of a sampling operator $\Phi : \mathcal{I} \to \mathbb{R}^{w\times h}$ evaluated at an image I, i.e. $z = \Phi(I)$ that is a vectorized version of the matrix Z with elements

$$Z_{ij} = I(i, j) \qquad i = 1, \dots, w; j = 1, \dots, h.$$

In order to counter the problem that the sampling operator is in general not injective, we add another constraint to $\mathcal{I}$ by requiring that Φ is bijective. Hence, the quantity $I_z = \Phi^{-1}(z)$ is well-defined. We denote by $\mathcal{X}$ the image of Φ. In the main text, we also refer to $\mathcal{T}(z, \Delta) = \Phi(g_\Delta(I_z)) \in \mathcal{X}$ as the image corresponding to the sampled image z transformed by the group element g_Δ.

We can now analogously define subsets in the finite dimensional vector space $G^z \subset \mathcal{X}$ by $G^z := \{\Phi(I) : I \in \mathbb{G}_\mathcal{S}(I_z)\}$. By definition of $\mathbb{G}_\mathcal{S}(I)$, the sets G^z are also disjoint. Any G^z with $\mathbb{P}(\{X \in G^z\}) \geq \mathbb{P}(\{X \in \mathbb{G}^z\})$ and $\mathbb{G}^z = \{\Phi(I) : I \in \mathbb{G}^{I_z}\}$ we call a *transformation set*.

B.6. Proof of Theorem 1

Please refer to Section B.5 for the necessary notation for this section. Furthermore, define $\mathcal{L}_{\text{adv}}(f) = \mathcal{L}_{rob}(f; 0, 0)$, $\mathcal{L}_{\text{nat}}(f) = \mathcal{L}_{\text{nat}}(f; 0, 0)$.

We prove the theorem by contradiction. Let us assume that $f^{\text{rob}} \notin \mathcal{V}$ for exactly one element $G^z \in \widetilde{\mathcal{G}}$ and the marginal distribution over $\widetilde{\mathcal{G}}$ that can be defined as $P(\{G^X = U\}) = P(\{X \in U\})$ for any $U \in \widetilde{\mathcal{G}}$, is discrete(for simplicity of presentation) and G^z has non-zero probability.

Let's assume that there is at least one invariant set G^z, on which f^{rob} is not constant and collect all different values in the set $A = \{f^{\text{rob}}(x) : x \in G^z\}$ (with cardinality strictly bigger than 1 since f not-constant) and denote the distribution over $x \in G^z$ by P_z. Since G^z is assumed to be a subset of a group orbit and $\mathbb{G}$-orbits partition $\mathcal{X}$, there is a mapping Ψ that maps each $x \in \mathcal{X}$ to a unique invariant set (see Section B.5). This property allows us to lower bound of the robust loss as follows for any $z \in \mathcal{X}$:

$$\begin{aligned}
&\mathbb{E}_{X,Y} \sup_{x' \in G^X} \ell(f(x'), Y) \\
=&\mathbb{E}\,[\sup_{x' \in G^X} \ell(f(x'), Y) | X \notin G^z]\mathbb{P}(\{X \notin G^z\}) + \mathbb{E}_{Y|z}[\sup_{x' \in G^z} \ell(f(x'), Y) | G^z]\mathbb{P}(\{X \in G^z\}) \\
\geq&\mathbb{E}\,[\sup_{x' \in G^X} \ell(f(x'), Y) | X \notin G^z]\mathbb{P}(\{X \notin G^z\}) + \sup_{a \in A} \int H(x, a) dP_z(x)
\end{aligned} \tag{5}$$

where the function H is specific to G^z and defined as $H(x, a) = \mathbb{E}_{Y|x}\ell(a, Y)$ and the inequality follows from

$$\mathbb{E}_{X|z}\mathbb{E}_{Y|x}[\sup_{a \in A} \ell(a, Y) | X = x] | G^z = \int \mathbb{E}_{Y|x} \sup_{a \in A} \ell(a, y) dP_z(x) \geq \sup_{a \in A} \int \mathbb{E}_{Y|x}\ell(a, Y) dP_z(x).$$

The right hand side is minimized with respect to the set A by choosing $A = \{a^\star\}$ with $a^\star = \arg\min_a \int H(x, a) dP_z(x)$ because setting $f(x) = a^\star$ for all $x \in G^z$ leads to equality in equation (5). Since $\mathbb{P}(\{X \in G^z\}) > 0$ by assumption, choosing $f^\star(x) = a^\star$ for all $x \in G^z$ implies $\mathcal{L}_{\text{adv}}(f^\star) < \mathcal{L}_{\text{adv}}(f^{\text{rob}})$ which contradicts optimality of f^{rob} and thus proves the first statement of the theorem.

For the second statement let us rewrite

$$\begin{aligned}
\mathcal{L}_{\text{adv}}(f) &= \mathcal{L}_{nat}(f) + [\mathcal{L}_{\text{adv}}(f) - \mathcal{L}_{nat}(f)] \\
&= \mathbb{E}\,\ell(f_\theta(\mathcal{T}(X, \Delta')), Y) + \underbrace{\mathbb{E}\,[\max_{\Delta' \in \mathcal{S}} \ell(f(\mathcal{T}(X, \Delta')), Y) - \ell(f(X), Y)]}_{\tilde{R}(f)}
\end{aligned}$$

By the first statement we know that the set

$$F^{\text{rob}} := \{f \in \mathcal{V} : \mathcal{L}_{\text{adv}}(f) \leq \mathcal{L}_{\text{adv}}(f') \quad \forall f' \in \mathcal{F}\}$$

is non-empty. For all $f \in F^{\text{rob}}$, it holds by definition of $\mathcal{V}$ that $\tilde{R}(f) = 0$.

Since $\mathcal{V}(R) \subset \mathcal{V}$, the minimizers $f^{\min}$ of (O1) satisfy $\mathcal{L}_{\text{nat}}(f^{\min}) \leq \mathcal{L}_{\text{nat}}(f)$ for all $f \in \mathcal{V}$. But since in $\mathcal{V}$ we have $\mathcal{L}_{\text{adv}}(f) = \mathcal{L}_{\text{nat}}(f)$ it directly follows that $f^{\min} \in F^{\text{rob}}$. The same argument goes through for (O2) since for all $f \in \mathcal{V}$, we have $\mathcal{L}_{\text{adv}}(f) = \mathcal{L}_{\text{nat}}(f)$. This concludes the proof of the theorem.

B.7. Proof of Theorem 2

Similar to the proof of Theorem 1, we can const minimizer $f^\star$ of the natural loss that is in $\mathcal{V}$ given the assumption that $Y \perp\!\!\!\perp X | G^z$. Hence since on $\mathcal{V}$ both losses are the same, together with Theorem 1 the robust minimizer also minimizes the unconstrained natural loss.

Assume $f^{\text{nat}} \notin \mathcal{V}$ minimizes $\mathcal{L}_{\text{nat}}(f)$, in particular, it is constant on all invariant sets except G^z for some $z \in \mathcal{X}$. Again by existence of a mapping Ψ and hence since by assumption, the joint conditional distribution $P_z(Y, X) = P_z(Y)P_z(X)$, we can write for any f

$$\begin{aligned}
\mathcal{L}_{\text{nat}}(f) &= \mathbb{E}_{X,Y}\ell(f(X), Y) \\
&= \mathbb{E}\left[\ell(f(X), Y) | X \notin G^z\right]\mathbb{P}(\{X \notin G^z\}) + \mathbb{E}\left[\ell(f(X), Y) | G^z\right]\mathbb{P}(\{X \in G^z\}) \\
&= \mathbb{E}\left[\ell(f(X), Y) | X \notin G^z\right]\mathbb{P}(\{X \notin G^z\}) + \mathbb{E}_X\mathbb{E}_Y\ell(f(X), Y)\mathbb{P}(\{X \in G^z\}).
\end{aligned}$$

Note that

$$\begin{aligned}
\mathbb{E}_X\mathbb{E}_Y\ell(f(X), Y)|G^z &= \int H(x)dP_z(x) \\
&\geq \int \min_{x \in G^z} H(x)dP_z(x) = \mathbb{E}_X\mathbb{E}_Y\ell(f^\star(X), Y)|G^z
\end{aligned} \tag{6}$$

Then, setting $f^\star(x) = \min_{x \in G^z} \mathbb{E}_Y\ell(f^{\text{nat}}(x), Y)|G^z$ for all $x \in G^z$ yields $\mathcal{L}_{\text{nat}}(f^\star) = \mathcal{L}_{\text{nat}}(f^{\text{nat}}) \leq \mathcal{L}_{\text{nat}}(f)$ for all $f \in \mathcal{F}$ by definition of f^{nat}.

If additionally the support of P_z is equal to G^z and ℓ is injective, the inequality (6) becomes a strict inequality for $f^{\text{nat}} \notin \mathcal{F}$ and hence we have $\mathcal{L}_{\text{nat}}(f^\star) < \mathcal{L}_{\text{nat}}(f^{\text{nat}})$ which contradicts the definition of f^{nat} being the minimizer of the natural loss.

C. Experimental details

C.1. Attacks and defenses

Grid search attack We consider a default grid of 5 values per translation direction and 31 values for rotation, yielding 775 transformed examples that are evaluated for each X_i. We refer to the accuracy attained under this attack as *grid accuracy*.[3]

Defenses **worst-of-k** In worst-of-k (Wo-k) adversarial training, at iteration t, we sample k different perturbations for each image in the batch. The one resulting in the highest function value is chosen to be the maximizer. Most of our experiments are conducted with $k = 10$ consistent with [10] as a higher k only improved performance minimally.

Spatial PGD. In analogy to common practice for ℓ_p adversarial training as in e.g. [34, 22], the S-PGD mechanism uses projected gradient descent with respect to the translation and rotation parameters with projection on the constrained set $\mathcal{S}$ of transformations. We consider 5 steps, starting from a random initialization, with step sizes of $[0.03, 0.03, 0.3]$ (following [10]) for horizontal-, vertical translation and rotation respectively.

Random. Data augmentation with a distinct random perturbation per image and iteration as ubiquitously used in practice. This can be seen as the most naive "adversarial" example as it corresponds to worst-of-k with $k = 1$.

Note that the empty space that results from translating and rotating an image is filled with black pixels (*constant padding*) if not otherwise noted, and reflections in some cases (see Section C.5). See Fig. 2 for an example.

[3] A finer grid search attack with a total of 7500 transformations only lead to minor reductions in accuracy compared to the coarse grid (see Table 6).

C.2. Spatial-equivariant networks

In our experiments, we report spatial robustness for the following spatial-equivariant networks: (a) G-ResNet44 (GRN) [7] using p4m convolutional layers (90 degree rotations, translations and mirror reflections) on CIFAR10; (b) Equivariant Transformer Networks (ETN) [35], a generalization of Polar Transformer Networks (PTN) [11], on SVHN; and (c) Spatial Transformer Networks (STN) [16] on SVHN. A more comprehensive discussion of previous literature on equivariant networks against translations and rotations can be found in Section 4. We choose the architectures listed above based on availability of reproducible code and previously reported state-of-the art accuracies for the respective datasets.

We train GRN, STN and ETN end-to-end using standard augmentation as described in Section **??** (std) and random rotations in addition (std*). In order to compute the grid accuracies, the adversarial attacks using grid search are then computed by evaluating the entire architecture. Out of curiosity we also trained a two-stage STN where we train the localization network separately on a supervised randomly transformed set of the training data to predict the transformation parameters treated as labels. Details about the implementation and results can be found in Section C.5.

We had difficulties to train both ETN and STN to higher than 86% natural accuracy for CIFAR10 even after an extensive learning rate and schedule search so we do not report the numbers here.

C.3. Training procedure

All models are trained using a single-GPU on a node equipped with an NVIDIA GeForce GTX 1080 Ti and two 10-core Xeon E5-2630v4 processors.

We run a standard minibatch SGD update with a momentum term with parameter 0.9 and weight decay parameter 0.0002. We use an initial learning rate of 0.1 which is divided by 10 after half and three-quarters of the training steps. Independent of the defense method, we fix the number of iterations to 80000 for SVHN and CIFAR-10, and to 120000 for CIFAR-100. For comparability across all methods, we fix the number of unique original images in each iteration to 64. For the baselines, we additionally trained with a batch size of 128 and report the higher accuracy.

C.4. Example images

One pair of example images is shown in panels (b) and (c) in Fig. 2, where the original image is depicted along with a transformed version $\mathcal{T}(\cdot, \Delta)$ with $\Delta \in \mathcal{S}$.

(a) (b) (c)

Figure 2. Example images and classifications by the Standard model. (a) An image that is misclassified for all rotations in the considered grid. (b) An image that is correctly classified for most of the rotations in the considered grid. (c) One rotation for which the image shown in (b) is misclassified as "airplane".

C.5. Design choices for two-stage STN

Since STNs are known to be sensitive to hyperparameter settings and thus difficult to train end-to-end [35], we apply the following two-stage procedure to simulate its functionality: (1) we first train a ResNet-32 as a localization regression network (LocNet) to predict the attack perturbation separately by learning from a training set, which contains perturbed images and uses the transformations as the prediction targets; (2) at the same time we train a ResNet-32 classifier with data augmentation, namely random translations and rotations; (3) during the test phase, the output of the LocNet is used by a spatial transformer module that transforms the image before entering the pretrained classifier. We refer to this two-stage STN as STN+.

For the classifiers, we take the two trained models for CIFAR10 and SVHN from our previous experiments for AT(rob, rnd). Since we do not expect the regressors to be perfect in terms of prediction capability, there will still be some transformation left after the regression stage. Thus, the classifiers should have a certain degree of robustness. Their training procedure is described in Section C.3.

The regressor is trained on zero padded – suffix (c) – as well as reflect padded inputs – suffix (r) – for comparison. The

former possibly yields an unfair advantage of this approach compared to other methods as the neural network can exploit the edges (induced through zero padding) to learn the transformation parameters. Therefore, we also consider reflection padding to assess the effect of the different paddings on final performance. Nonetheless, zero padding is consistent with the augmentation setting for the end-to-end trained networks and regularized methods and was also the choice considered by [10]. For completeness we also show results when using reflection padding for training LocNet although it lacks comparability with the other methods since attacks should be reflection-padded as well.

[10] additionally analyze a "black canvas" setting where the images are padded with zeros prior to applying the transformation, ensuring that no information is lost due to cropping. Their experiments show that the reduced accuracy of the models cannot be attributed to this effect. Since both versions yield similar results, we report results on the first version of pad and crop choices, having input images of the same size as the original.

In the spatial transformer module we compares two variants of handling the labels predicted by the regressor. We can either tilt the transformed image with the opposite transformation of the predicted labels, which will, under the assumption that the regressor successfully learnt object orientations, turn back the image but potentially result in extra padding space before we feed the images into the classifier. Alternatively, we can subtract the predicted transformation from the attack transformation, then use the remaining transformation as the new "attack transformation". The latter will result in much smaller padding areas, if the regressor is performing well. From the experimental results we do see a big drop if we naively transform images twice. We denote the former method as "naive" and latter as "trick".

For CIFAR-10, this two-stage classifier achieved relatively high grid accuracies. However, the obtained accuracies are still lower than expected, given that the regressor is allowed to learn rotations with a separately trained regressor on the transformed training set. For SVHN we also see a gain compared to adversarial training without regularizer. However, the performance still lags behind the accuracies obtained by the regularizers. The results are summarized in Table 3.

Table 3. Accuracies of two-stage STN (STN+) under different settings. Details are provided in Section C.5.

Dataset	STN+(c) trick	STN+(r) trick	STN+(c) naive	STN+(r) naive
SVHN (nat)	94.92	95.51	94.92	95.51
(grid)	90.95	90.28	64.91	59.68
CIFAR10 (nat)	91.29	90.99	91.29	90.99
(grid)	83.05	84.31	44.88	42.84

C.6. Additional experimental results

In this section we discuss additional experimental results we collected and analyses we performed.

Weakness of first order attack. Table 5 shows the accuracies of various models trained with S-PGD defenses and evaluated against the S-PGD and the grid search attack on all datasets. We observe that the S-PGD attack constitutes are very weak attack since the associated accuracies are much larger than for the grid search attack. In other words, the S-PGD attack only yields a very loose upper bound on the adversarial accuracy. This stands in stark contrast to ℓ_∞ attacks and has first been noted and discussed in [10]. Interestingly, using the first order method as a *defense* mechanism proves to be very effective in terms of grid accuracy. When used in combination with a regularizer this defense yields the largest overall accuracies as shown and discussed in Section 3. Recall that due to computational reasons grid search cannot be used as a defense mechanism. Therefore, the strongest computationally feasible defense does not use the same mechanism as the strongest attack in our setting.

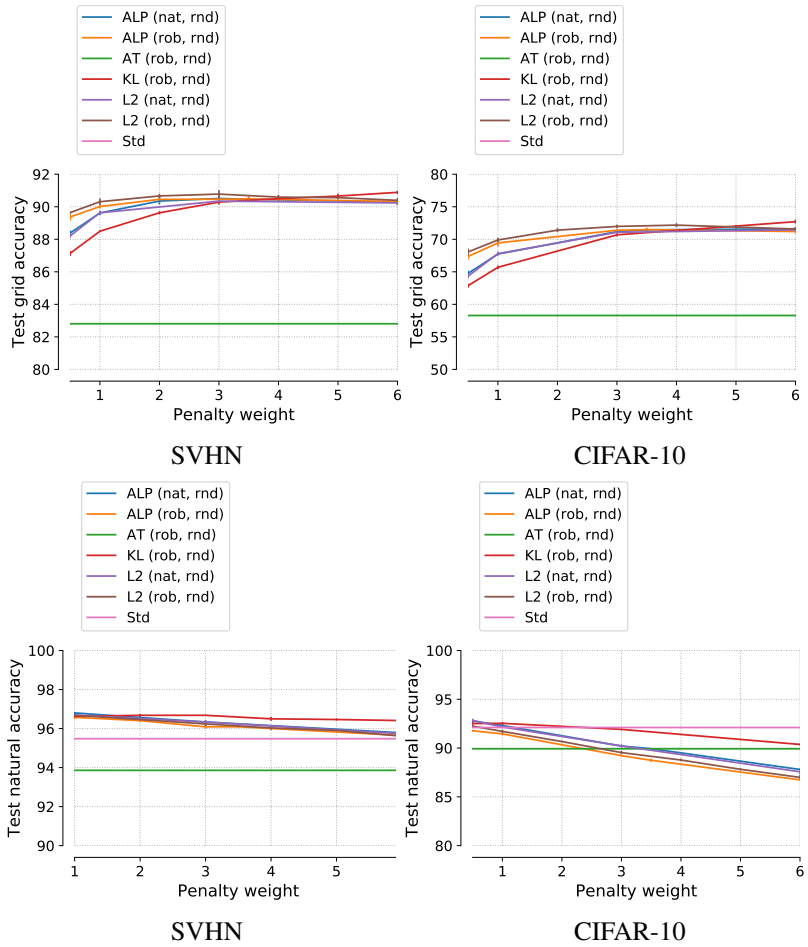

Figure 3. Test grid accuracy (first row) and test natural accuracy (second row) as a function of the regularization parameter λ for the SVHN (first column) and CIFAR-10 (second column) datasets and data augmentation ("rnd"). The test grid accuracy is relatively robust in a large range of λ values while natural test accuracy decreases with larger values of λ.

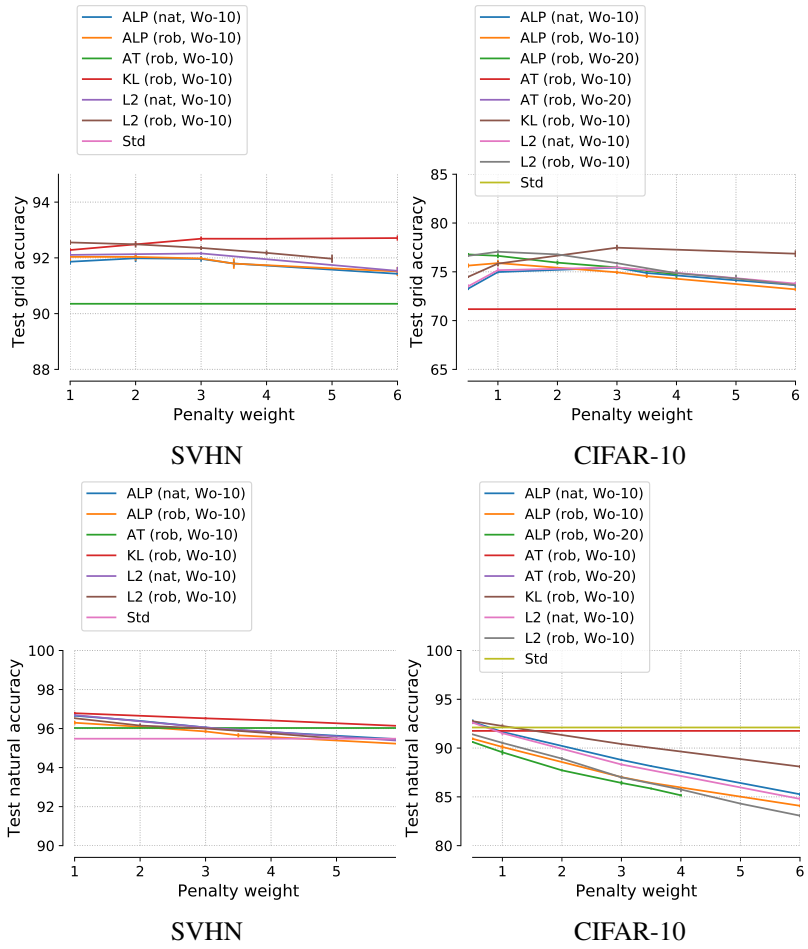

Figure 4. Test grid accuracy (first row) and test natural accuracy (second row) as a function of the regularization parameter λ for the SVHN (first column) and CIFAR-10 (second column) datasets and Wo-k defenses. The test grid accuracy is relatively robust in a large range of λ values while natural test accuracy decreases with larger values of λ.

Table 4. Mean accuracies of different models trained with and without defenses and evaluated on the natural test set (column "Natural") and against the grid search attack (column "Grid") on CIFAR-100. Standard errors are shown in parentheses.

	Training \ Evaluation	Natural	Grid
CIFAR-100	std	70.23% (0.18%)	5.09% (0.25%)
	AT(mix, rnd)	72.38% (0.17%)	23.43% (0.47%)
	AT(rob, rnd) $\equiv$ std*	66.62% (0.37%)	28.53% (0.25%)
	ALP(mix, rnd) ($\lambda = 0.75$)	66.54% (0.26%)	**45.97% (0.22%)**
	AT(mix, Wo-10)	73.03% (0.13%)	35.93% (0.24%)
	AT(rob, Wo-10)	68.79% (0.34%)	38.21% (0.10%)
	ALP(mix, Wo-10) ($\lambda = 0.25$)	68.54% (0.27%)	**49.30% (0.33%)**
	AT(mix, Wo-20)	73.33% (0.22%)	38.06% (0.19%)
	AT(rob, Wo-20)	69.15% (0.49%)	40.09% (0.31%)
	ALP(mix, Wo-20) ($\lambda = 0.25$)	68.04% (0.27%)	**49.98% (0.31%)**
	AT(mix, S-PGD)	71.11% (0.37%)	33.40% (0.21%)
	AT(rob, S-PGD)	68.87% (0.19%)	37.87% (0.12%)
	ALP(mix, S-PGD) ($\lambda = 0.5$)	68.44% (0.39%)	**52.58% (0.20%)**

Table 5. Mean accuracies of different models trained with S-PGD defenses and evaluated on the natural test set (column "Natural"), against the S-PGD attack (column "S-PGD") and against the grid search attack (column "Grid") on the SVHN, CIFAR-10 and CIFAR-100 datasets. While the test accuracy for the S-PGD attack is only slightly lower than the natural accuracy in most cases, the grid accuracy is significantly smaller. In other words, the S-PGD attack only yields a loose upper bound on adversarial accuracy while it is a competitive defense in terms of test grid accuracy when used together with ALP.

	Training \ Evaluation	Natural	S-PGD	Grid
SVHN	AT(mix, S-PGD)	96.27% (0.00%)	95.26% (0.04%)	84.81% (0.01%)
	AT(rob, S-PGD)	96.06% (0.10%)	95.46% (0.10%)	87.29% (0.09%)
	ALP(mix, S-PGD) ($\lambda = 2.5$)	96.30% (0.09%)	95.92% (0.13%)	**92.42% (0.20%)**
CIFAR-10	AT(mix, S-PGD)	92.19% (0.23%)	88.84% (0.27%)	64.26% (0.25%)
	AT(rob, S-PGD)	91.83% (0.19%)	89.87% (0.10%)	69.74% (0.27%)
	ALP(mix, S-PGD) ($\lambda = 3$)	89.70% (0.10%)	88.15% (0.21%)	**77.72% (0.35%)**
CIFAR-100	AT(mix, S-PGD)	71.11% (0.37%)	65.01% (0.32%)	33.40% (0.21%)
	AT(rob, S-PGD)	68.87% (0.19%)	65.56% (0.12%)	37.87% (0.12%)
	ALP(mix, S-PGD) ($\lambda = 0.5$)	68.44% (0.39%)	66.04% (0.40%)	**52.58% (0.20%)**

Table 6. Mean accuracies for different models evaluated against two different grid search attacks. The column "Grid (775)" shows the test grid accuracy using the default grid containing 5 values per translation direction and 31 values for rotation, yielding a total of 775 transformed examples that are evaluated for each X_i. The column "Grid (7500)" shows the test grid accuracy on a much finer grid with 10 values per translation direction and 75 values for rotation, resulting 7500 transformed examples. We observe that the test grid accuracy only decreases slightly for the finer grid and the reduction in accuracy is smaller for ALP than for AT. Due to computational reasons we use the grid containing 775 values for all other experiments.

	Training \ Evaluation	Grid (775)	Grid (7500)
SVHN	$\text{AT}(\text{mix}, \text{Wo-10})$	88.83% (0.10%)	88.02% (0.12%)
	$\text{AT}(\text{rob}, \text{Wo-10})$	89.75% (0.17%)	89.29% (0.15%)
	$\text{ALP}(\text{mix}, \text{Wo-10})$ $(\lambda = 1.5)$	92.17% (0.11%)	91.79% (0.12%)
CIFAR-10	$\text{AT}(\text{mix}, \text{Wo-10})$	68.14% (0.48%)	65.69% (0.28%)
	$\text{AT}(\text{rob}, \text{Wo-10})$	70.35% (0.16%)	68.28%(0.16%)
	$\text{ALP}(\text{mix}, \text{Wo-10})$ $(\lambda = 1)$	75.89% (0.23%)	74.58% (0.16%)
CIFAR-100	$\text{AT}(\text{mix}, \text{Wo-10})$	35.93% (0.24%)	33.62% (0.23%)
	$\text{AT}(\text{rob}, \text{Wo-10})$	38.21% (0.10%)	36.04% (0.21%)
	$\text{ALP}(\text{mix}, \text{Wo-10})$ $(\lambda = 0.25)$	49.30% (0.33%)	47.95% (0.23%)

Table 7. Mean accuracies of models trained *without* regularized adversarial training. Standard errors are shown in parentheses.

	std	std*	AT(rob, Wo-10)
SVHN (nat)	95.48 (0.15)	93.97 (0.09)	96.03 (0.04)
(grid)	18.85 (1.27)	82.60 (0.23)	90.35 (0.27)
CIFAR10 (nat)	92.11 (0.18)	89.93 (0.18)	91.76 (0.23)
(grid)	9.52 (0.66)	58.29 (0.60)	71.17 (0.26)

Table 8. Mean accuracies of models trained with various forms of regularized adversarial training, using the KL regularization function. Standard errors are shown in parentheses.

	KL(nat, rnd)	KL(nat, Wo-10)	KL(rob, Wo-10)	KL-C(mix, S-PGD)	KL(nat, S-PGD)
SVHN (nat)	96.16 (0.10)	96.00 (0.02)	96.13 (0.07)	96.14 (0.04)	96.54 (0.01)
(grid)	90.69 (0.05)	92.27 (0.09)	92.71 (0.09)	92.42 (0.03)	92.62 (0.03)
CIFAR10 (nat)	89.33 (0.16)	90.83 (0.18)	90.41 (0.05)	89.98 (0.21)	89.82 (0.13)
(grid)	73.50 (0.19)	77.34 (0.19)	77.47 (0.28)	78.93 (0.23)	78.89 (0.07)

Table 9. Mean accuracies of models trained with various forms of regularized adversarial training, using the ℓ_2 and ALP regularization functions. Standard errors are shown in parentheses.

	ℓ_2(nat, Wo-10)	ℓ_2(adv, Wo-10)	ALP(mix, Wo-10)	ALP(rob, Wo-10)	ALP(rob, S-PGD)
SVHN (nat)	96.05 (0.04)	96.53 (0.03)	96.41 (0.07)	96.3 (0.09)	96.11 (0.08)
(grid)	92.16 (0.05)	92.55 (0.08)	92.17 (0.11)	92.04 (0.19)	92.32 (0.17)
CIFAR10 (nat)	88.32 (0.13)	90.53 (0.16)	91.13 (0.13)	90.11 (0.25)	89.85 (0.27)
(grid)	75.46 (0.25)	77.06 (0.16)	75.89 (0.23)	75.90 (0.31)	77.80 (0.17)

Table 10. Mean accuracies of models trained with various forms of augmented training, i.e. unregularized and regularized data augmentation. Standard errors are shown in parentheses.

	std*	ℓ_2(nat, rnd)	KL(nat, rnd)	ALP(rob, rnd)	KL(rob, rnd)	ALP(mix, rnd)
SVHN (nat)	93.97 (0.09)	96.34 (0.08)	96.16 (0.10)	96.09 (0.06)	96.23 (0.08)	96.19 (0.07)
(grid)	82.60 (0.23)	90.51 (0.15)	90.69 (0.05)	90.48 (0.16)	90.92 (0.17)	90.48 (0.15)
CIFAR10 (nat)	89.93 (0.18)	87.80 (0.11)	89.34 (0.16)	88.75 (0.18)	89.47 (0.04)	89.43 (0.28)
(grid)	58.29 (0.60)	71.60 (0.27)	73.50 (0.19)	71.49 (0.30)	73.22 (0.14)	71.97 (0.11)

Table 11. Mean accuracies of models trained with various forms of regularized adversarial training. Standard errors are shown in parentheses.

	ALP(nat, Wo-10)	ℓ_2(nat, Wo-10)	KL-C(nat, Wo-10)	KL(nat, Wo-10)
SVHN (nat)	96.39 (0.03)	96.05 (0.04)	96.18 (0.06)	96.00 (0.02)
(grid)	91.98 (0.13)	92.16 (0.05)	91.99 (0.12)	92.27 (0.09)
CIFAR10 (nat)	88.78 (0.11)	88.32 (0.13)	89.61 (0.09)	90.83 (0.18)
(grid)	75.43 (0.13)	75.46 (0.25)	76.15 (0.23)	77.34 (0.19)

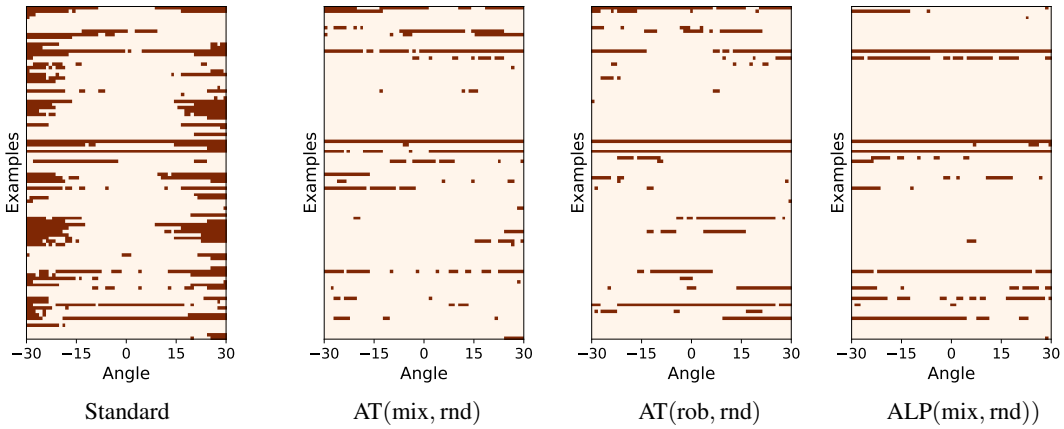

Figure 5. For 100 randomly chosen examples from the CIFAR-10 dataset, we show which rotations lead to a misclassification by various models trained with and without data augmentation ("rnd"). Each row corresponds to one example and each column to one angle. A dark red square indicates that the corresponding example was misclassified after being rotated by the corresponding angle. The visualization for ALP(mix, rnd) is more fragmented than for ALP(rob, rnd) and ALP(mix, rnd).

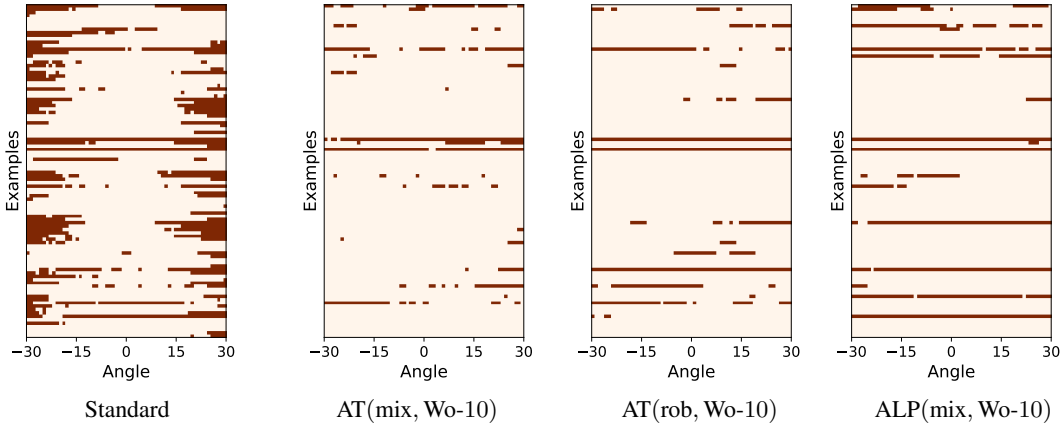

Figure 6. For 100 randomly chosen examples from the CIFAR-10 dataset, we show which rotations lead to a misclassification by various models trained with and without the Wo-10 defense. Each row corresponds to one example and each column to one angle. A dark red square indicates that the corresponding example was misclassified after being rotated by the corresponding angle. The visualization for AT(mix, Wo-10) is more fragmented than for AT(rob, Wo-10) and ALP(mix, Wo-10).