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

Fanny Yang [1] [2] **Zuowen Wang** [3] **Christina Heinze-Deml** [4]

Abstract

This work provides theoretical and empirical evidence that invariance-inducing regularizers can increase predictive accuracy for worst-case spatial transformations (spatial *robustness*). Evaluated on these *adversarially* transformed examples, we demonstrate that adding regularization on top of standard or adversarial training reduces the relative error by 20% for CIFAR10 without increasing the computational cost. This outperforms handcrafted networks that were explicitly designed to be spatial-equivariant. Furthermore, we observe for SVHN, known to have inherent variance in orientation, that robust training also improves standard accuracy on the test set. We prove that this no-trade-off phenomenon holds for adversarial examples from *transformation* groups in the infinite data limit.

1. Introduction

As deployment of machine learning systems in the real world has steadily increased over recent years, the trustworthiness of these systems is of crucial importance. 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 neural networks have been shown to be expressive both theoretically [17, 4, 14] and empirically [38], in this work we study to what extent standard neural networks predictors can be made invariant to small rotations and translations. In contrast to enforcing conventional invariance on entire group orbits, we weaken the goal to invariance on smaller so-called *transformation sets*. During test time we

assess transformation set invariance by computing the prediction accuracy on the worst-case (*adversarial*) transformation in the (small) transformation set of each image in the test data. The higher this worst-case prediction accuracy of a model is, the more *spatially robust* we say it is.

Recently, it was observed in [10, 12, 29, 19, 13, 2] that worst-case prediction performance drops dramatically for neural network classifiers obtained using standard training, even for rather small transformation sets. In this context, we examine the effectiveness of regularization that explicitly encourages the predictor to be constant for transformed versions of the same image. Broadly speaking, there are two approaches to encourage invariance of neural network predictors. On the one hand, the relative simplicity of the mathematical model for rotations and translations has led to carefully hand-engineered architectures that incorporate spatial invariance directly [18, 22, 7, 25, 35, 34, 11, 32]. On the other hand, augmentation-based methods [3, 37] constitute an alternative approach to encourage desired invariances on transformation sets. Specifically, the idea is to augment the training data by a random or smartly chosen transformation of every image for which the predictor output is enforced to be close to the output of the original image. This *invariance-inducing* regularization term is then added to the cross entropy loss for back-propagation.

In the empirical sections of this paper, we hence want to explore the following questions: (a) To what extent can augmentation and regularization based methods improve spatial robustness of common deep neural networks? (b) How does augmentation-based invariance-inducing regularization perform in case of small spatial transformations compared to representative specialized architectures designed to achieve invariance against entire transformation groups?

As a justification for employing this form of *invariance-inducing* regularization, we prove in our theoretical section in Appendix B that when perturbations come from transformation groups, predictors that optimize the robust loss are in fact invariant on the set of transformed images. Although recent works show a fundamental trade-off between robust and standard accuracy in constructed ℓ_p perturbation settings [33, 39, 30], we additionally show that this is fundamentally different for spatial transformations due to their

[1]Department of Statistics, Stanford [2]Instiue of Theoretical Studies, ETH Zurich [3]Department of Computer Science, ETH Zurich [4]Seminar for Statistics, ETH Zurich. Correspondence to: F. Yang, C. Heinze-Deml <fannyang, heinzedeml@stat.math.ethz.ch>.

group structure.

For the empirical study, we compare spatial robustness for augmentation-based methods and specialized neural network architectures on CIFAR-10 and SVHN. 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. Furthermore, they outperform representative handcrated networks that were explicitly designed for invariance.

2. Experimental setup

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

2.1. Regularization based methods

Denoting by G^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

- $R_{\ell_2}(f, X, Y) = \sup_{x' \in G^X} \|f(X) - f(x')\|_2^2$

- $R_{\text{KL}}(f, X, Y) = \sup_{x' \in G^X} \mathrm{D}_{\text{KL}}(f(x'), f(X))$ (same as [39] 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[20])

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]. 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: (a) Wo-k: the worst of k randomly drawn transformations; (b) S-PGD: Several steps of Projected Gradient Descent with respect to the spatial transformation parameters; (c) 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 we use a naming convention consisting of the following components: (a) `Reg`: refers to what regularizer was used (ALP, ℓ_2, KL or AT) where AT refers 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

In this section we assess the invariance against small translations and rotations of estimators obtained by standard, regularized (adversarial) training techniques and specialized spatial equivariant architectures. In order to measure invariance, we evaluate the *natural test accuracy* (standard accuracy on the test set) and robust accuracy

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

where G^X are sets of small spatial transformations of X, defined rigorously in Section B, and the expectation is the empirical distribution of the test set. Unless otherwise noted, the inner maximization is implemented using grid search (as defined in Section C.1).

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 the Appendix, Tables 4–9. For

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). ⋆ 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	–	–	–	–

each method, we report averages (standard errors are contained in Tables 4–9) computed over five training runs with identical hyperparameter settings. We compare all methods by computing absolute and relative error reductions (defined as $\frac{\text{absolute error drop}}{\text{prior error}}$).

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 [21]. For the regularized methods, we train a ResNet-32 [15], 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 left-right 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 regularizers 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) (green line) and ALP(rob, Wo-k) (red line) 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? While the leap in robust accuracy from Wo-1 (also referred to as rnd) to Wo-10 is quite large, increasing k to 20 only gives diminishing returns while requiring $\sim 3\times$ more training time. Furthermore, for any fixed training time, regularized methods ex-

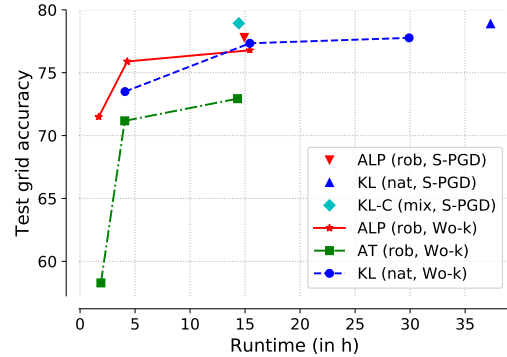hibit higher robust accuracies where the gap varies with the particular choice of regularizer and defense mechanism.

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

Comparison with spatial equivariant networks Although the rotation-augmented G-ResNet44 obtains higher grid (SVHN: 84.9%, CIFAR-10: 71.64%) and natural accuracies (SVHN: 95%, CIFAR-10: 93.08%) than the rotation-augmented Resnet-32 on both SVHN (grid: 82.60%, nat: 93.97%) and CIFAR10 (grid: 58.29%, nat: 89.93%), regularizing standard data 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 (ℓ_2: 71.60%, KL: 73.50%) and surpasses it on SVHN on grid (ℓ_2: 90.51%, KL: 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.[1] In conclusion, regularized augmentation based methods match or outperform representative end-to-end networks handcrafted to be equivariant to spatial transformations.

---

[1] 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.

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

Trade-off natural vs. adversarial accuracy SVHN is one of the main datasets (without artificial augmentation like in rot-MNIST [23]) 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 in Theorem 2 of the distribution on the transformation sets having support G^z. Furthermore, as all numbers in SVHN have the same label irrespective of small rotations of at most 30 degrees, the first assumption in Theorem 2 is also fulfilled. Table 1 and 2 confirm the statement in the Theorem that improving robust accuracy may not hurt natural accuracy or even improve it: For SVHN, adding regularization to samples obtained both via Wo-10 adversarial search or random transformation (rnd) consistently not only helps robust but also standard accuracy.

Comparing the effects of different regularization parameters on test grid accuracy We study Tables 1 and 2 and attempt to disentangle the effects by varying only one parameter. For example we can observe that fixing any regularizer defense to Wo-10, the robust regularized loss Reg (rob, Wo-10) (i.e., $\mathcal{L}_{\mathrm{rob}}(f; R)$) does better (or not statistically significantly worse) than Reg (nat, Wo-10) (i.e., $\mathcal{L}_{\mathrm{nat}}(f; R)$). Furthermore, the KL regularizer generally performs better than ℓ_2 for a large number of settings. A possible explanation for the latter could be that D_{KL} upper bounds the squared ℓ_2 loss on the probability simplex and is hence more restrictive.

Choice of λ The different regularization methods peak at different λ in terms of grid accuracy. 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 5 in the Appendix.

Many more insightful experiments can be found in Section C.8.

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 [18] 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 [22, 25, 7, 41, 35]. For continuous transformations, steerability [34, 8] and coordinate transformation [11, 32] 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 [16], domain generalization [26], ℓ_p adversarial training [27, 20, 39], robustness against simple transformations [6] and semi-supervised learning [40, 36]. These approaches are based on augmenting the training data either *statically* [16, 26, 6, 36], ie. before fitting the model, or *adaptively* in the sense of adversarial training, with different augmented examples per training image generated in every iteration [20, 27, 39].

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 [19]) and simple random or grid search [10, 29]. 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 settings.

6. Acknowledgements

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] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *arXiv preprint arXiv:1811.11553*, 2018.

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

[4] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Info. Theory*, 39(3):930–945, 1993.

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

[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 *Proceedings of the International Conference on Machine Learning*, pages 2990–2999, 2016.

[8] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *Proceedings of the 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 *Proceedings of the International Conference on Machine Learning*, 2019.

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

[12] A. Fawzi and P. Frossard. Manitest: Are classifiers really invariant? In *British Machine Vision Conference (BMVC)*, 2015.

[13] Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 7549–7561, 2018.

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

[15] 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 Patern Recognition*, pages 770–778, 2016.

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

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

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[19] 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 Patern Recognition*, pages 4441–4449, 2018.

[20] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial Logit Pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 4, University of Toronto, 2009.

[22] 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 Patern Recognition*, pages 289–297, 2016.

[23] 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, 2007.

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

[25] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5058–5067, 2017.

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

[27] Taesik Na, Jong Hwan Ko, and Saibal Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. In *Proceedings of the 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] 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.

[30] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*, 2019.

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

[32] Kai Sheng Tai, Peter Bailis, and Gregory Valiant. Equivariant Transformer Networks. In *Proceedings of the International Conference on Machine Learning*, 2019.

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

[34] Maurice Weiler, Fred A Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant CNNs. In *Proceedings of the IEEE Conference on Computer Vision and Patern Recognition*, 2018.

[35] 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 Patern Recognition*, pages 5028–5037, 2017.

[36] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

[37] Larry S. Yaeger, Richard F. Lyon, and Brandyn J. Webb. Effective training of a neural network character classifier for word recognition. In *Advances in Neural Information Processing Systems*, pages 807–816, 1997.

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

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

[40] 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 Patern Recognition*, pages 4480–4488, 2016.

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

A. Supplementary material

B. Theoretical results for invariance-inducing regularization

In this section, we first introduce our notion of transformation sets and formalize robustness against a small range of translations and rotations. We then prove that, on a population level, constraining or regularizing for transformation set invariance yields models that minimize the robust loss. Moreover, when the label distribution is constant on each transformation set, we show that 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, $X \in \mathcal{X} \subset \mathbb{R}^d$ are the observed images, $Y \in \mathbb{R}^p$ is 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 vector that is then used for prediction via a softmax layer.

B.1. Transformation sets

Invariance with respect to 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 $z \in \mathcal{X}$. G^z is called a *transformation set*. For example in the case of rotations, the transformation set G^z corresponds to the set of observed images in a dataset that are different versions of the same image z, that can be obtained by small rotations of one another.

By the technical assumption on the space of real images that the sampling operator is bijective, the mapping $z \to G^z$ is 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. Importantly, the bijectivity assumption also leads to G^z being disjoint for different images $z \in \mathcal{X}$. The above definition is distribution dependent and $\mathcal{G}$ partitions the support $\widetilde{\mathcal{X}}$ of the distribution. More details on the aforementioned concepts and definitions can be found in Sec. B.5 in the Appendix.

We say that a function f is *(transformation-)invariant* if $f(x) = f(x')$ for all $x, x' \in U$ for all $U \in \mathcal{G}$ and denote the class of all such functions by $\mathcal{V}$. Using this notation, fitting a model with high accuracy under worst-case "small" transformations of the input can be mathematically captured by the robust optimization formulation [5] of minimizing the *robust loss*

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

in some function space $\mathcal{F}$. We call the solution of this problem the (spatially) *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 optimization landscapes encountered when training neural networks. Furthermore, we show in the following section that for robustness over transformation sets, constraining the model class to invariant functions leads to the same optimizer of the robust loss. These facts motivate invariance-inducing regularization which we then show to exhibit improved robust test accuracy in practice.

B.2. Regularization to encourage invariance

For any regularizer R, we define the corresponding constrained set of functions $\mathcal{V}(R)$ as

$$\mathcal{V}(R) := \{f : R(f, x, y) = 0 \quad \forall (x, y) \in \text{supp}(\mathbb{P})\},$$

where $\text{supp}(\mathbb{P})$ denotes the support of $\mathbb{P}$. When $R(f, x, y) = \sup_{x' \in G^x} h(f(x), f'(x))$ and h is a semimetric[2] on $\mathbb{R}^p$, we have $\mathcal{V}(R) = \mathcal{V}$. We now consider constrained optimization problems of the form

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

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

---

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

The following theorem shows that (O1), (O2) are equivalent to (1) if the set of all invariant functions $\mathcal{V}$ is a subset of the function space $\mathcal{F}$.

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

The proof of Theorem 1 can be found in the Appendix in Sec. B.6. Since exact projection onto the constrained set is in general not achievable for neural networks, an alternative method to induce invariance is to relax the constraints by only requiring $f \in \{f : R(f, x, y) \leq \epsilon \quad \forall (x, y) \in \mathrm{supp}(\mathbb{P})\}$. Using Lagrangian duality, (O1) and (O2) can then be rewritten in penalized form for some scalar $\lambda > 0$ as

$$\min_{f \in \mathcal{F}} \mathcal{L}_{nat}(f; R, \lambda) := \min_{f \in \mathcal{F}} \mathbb{E}\, \ell(f(X), Y) + \lambda R(f, X, Y), \tag{2}$$

$$\min_{f \in \mathcal{F}} \mathcal{L}_{rob}(f; R, \lambda) := \min_{f \in \mathcal{F}} \mathbb{E}\, \sup_{x' \in G^X} \ell(f(x'), Y) + \lambda R(f, X, Y). \tag{3}$$

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 Eqs. (2) and (3). On the other hand, the constrained regularization formulation corresponds to restricting the function space and is hence comparable with hand-crafted network architecture design as described in Sec. C.2.

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

Even though high 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 $\mathcal{L}_{\mathrm{nat}}(f) := \mathbb{E}_{X,Y} \ell(f(X), Y)$ [33, 39]. In this section we address this question and identify the conditions for transformation set perturbations under which minimizing the robust loss does not lead to decreased natural accuracy. Notably, it even increases under mild assumptions.

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, it translates into the conditional distribution Y given x, denoted by $\mathbb{P}_{G^x}$, being constant for all x belonging to the same subset $U \in \mathcal{G}$. In other words, Y is conditionally independent of X given G^X, i.e. $Y \perp\!\!\!\perp X | G^X$. Under this assumption the next theorem shows 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, $\mathbb{P}_{G^z}$ has support G^z for every $z \in \widetilde{\mathcal{X}}$ 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 minimize standard test error. Practically, the assumption on the distribution of the transformation sets G^z corresponds to assuming non-zero inherent transformation variance in the natural distribution of the dataset. In practice, we indeed 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 [33, 39] that presented toy examples for which the ℓ_∞ robust solution must have higher natural loss than the Bayes optimal solution even in the infinite data limit. On a fundamental level, ℓ_∞ perturbation sets are of different nature compared to transformation sets on generic distributions of $\mathcal{X}$. In the distribution considered in [33, 39], there is no unique mapping from $x \in \mathcal{X}$ to a perturbation set and thus the conditional independence property does not hold in general.

B.4. Different regularizers and practical implementation

In order to improve robustness against spatial transformations we consider different choices of $R(f, x, y)$ in the regularized objectives (2) and (3) that we then compare empirically in Sec. 3. This allows us to view a number of variants of adversarial training in a unified framework. Broadly speaking, each approach listed below consists of first searching an adversarial example according to some mechanism which is then included in a *regularizing function*, often some weak notion of

distance between the prediction at X and the new example. The following choices of regularizers involve the maximization of a regularizing function over the transformation set

$$R_{\text{AT}}(f, X, Y) = \sup_{x' \in G^X} \ell(f(x'), Y) - \ell(f(X), Y) \text{ (equivalent to [31, 24] for } \mathcal{L}_{\text{nat}})$$

$$R_{\ell_2}(f, X, Y) = \sup_{x' \in G^X} \|f(X) - f(x')\|_2^2$$

$$R_{\text{KL}}(f, X, Y) = \sup_{x' \in G^X} D_{\text{KL}}(f(x'), f(X)) \text{ (equivalent to [39] for } \mathcal{L}_{\text{nat}})^3$$

where D_{KL} is the KL divergence on the softmax of the (logit) vectors $f \in \mathbb{R}^p$. In all cases we refer to the maximizer as an *adversarial example* that is found using *defense mechanisms* as discussed in Section 2.2. Note that for R_{ℓ_2} and R_{KL} the assumption $\mathcal{V}(R) \subseteq \mathcal{V}$ in Theorem 1 is satisfied.

Instead of performing a maximization of the regularizing function to find the adversarial example x', we can also choose x' in alternative ways The following variants are explored in the paper, two of which are reminiscent of previous work

$$R_{\text{ALP}}(f, X, Y) = \|f(x') - f(X)\|_2^2 \quad \text{with} \quad x' = \arg\max_{u \in G^X} \ell(f(u), Y) \text{ (equivalent to [20])}$$

$$R_{\text{KL-C}}(f, X, Y) = D_{\text{KL}}(f(x'), f(X)) \quad \text{with} \quad x' = \arg\max_{u \in G^X} \ell(f(u), Y)$$

$$R_{h-DA}(f, X) = \mathbb{E}_{x' \in G^X} h(f, X, X') \text{ (similar to [16])}$$

The last regularizer suggests using an additive penalty on top of data augmentation, with either one or even multiple random draws, where the penalty can be any of the above semimetrics h between $f(X)$ and $f(x')$, such as the ℓ_2 or D_{KL} distance. Albeit suboptimal, the experimental results in Section 3 suggest that simply adding the additive regularization penalty on top of randomly drawn data matches general adversarial training in terms of robust prediction at a fraction of the computational cost. In addition, Theorem 2 suggests that even when the goal is to *improve standard accuracy* and one expects inherent variance of nuisance factors in the data distribution it is likely helpful to use regularized data augmentation with R_{h-DA} instead of vanilla data augmentation. Empirically we observe this on the SVHN dataset in Section 3.

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

$$\min_{f \in \mathcal{F}} \mathbb{E} \sup_{\Delta \in \mathcal{S}} \ell(f(\mathcal{T}(X, \Delta)), Y) \geq \min_{f \in \mathcal{F}} \mathbb{E} \sup_{x' \in G^X} \ell(f(x'), Y) \tag{4}$$

where $\mathcal{S}$ is the set of transformations that we search over and $\mathcal{T}(X, \Delta)$ denotes the transformed image with transformation Δ (see Sec. B.5 in the Appendix for an explicit construction of the transformation search set $\mathcal{S}$). The left hand side in (4) 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.

In Figure 3 one pair of example images is shown: the original image (panel (a)) is depicted along with a transformed version $\mathcal{T}(\cdot, \Delta)$ with $\Delta \in \mathcal{S}$ (panel (b)) and the respective predictions by a standard neural network classifier.

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

In the following we introduce the concepts that are needed to rigorously define transformation sets that are subsets of the finite-dimensional (sampled) image space $\mathcal{X} \subset \mathbb{R}^d$. In particular, because rotations of continuous angles are not well-defined for sampled images we need to introduce the space of image functions $\mathcal{I}$ with elements $I : \mathbb{R}^2 \to [0, 255]^3$, i.e. I maps Euclidean coordinates in $\mathbb{R}^2$ to the RGB intensities of an image. The observed finite-dimensional vector is then a sampled version of an image function I. Here we assume that the sampling operator is bijective, with rigorous definitions later in the section.

Next we define subsets in the continuous function space and then transfer the concept back to the finite-dimensional $\mathcal{X}$. Let us define the symmetric group $\mathbb{G}$ of all rotations and horizontal and vertical translations acting on $\mathcal{I}$. We denote the elements in the group by g_Δ, uniquely parameterized by $\Delta \in \mathbb{R}^3$ and can be represented by a coordinate transform matrix G_Δ, see e.g. [7]. Two of the three dimensions represent the values for the translations and the third represents the rotation.

The transformed image (function) $g_\Delta(I) \in \mathcal{I}$ can be expressed by $g_\Delta(I)(v) = I(G_\Delta^{-1}v)$ 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 $\mathbb{G}(I) := \{g_\Delta(I) : g_\Delta \in \mathbb{G}\}$. By definition, the group orbits partition the space $\mathcal{I}$ and every $I \in \mathcal{I}$ belongs to a unique orbit.

Subsets of orbits In our 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 no longer fulfilled. For this purpose we consider a closed subset $\mathbb{G}^I$ of each group orbit. It follows from the group orbit definition that for every I it either belongs to one unique or no such set.

As described in the paragraph of Equation (4), when observing a (sampled) image I' in the training set, we do not know where in its corresponding subset $\mathbb{G}^{I'}$ it lies. At the same time, for our augmentation-based methods, we do not want the set $\mathcal{S}$ of transformations that we search over (*transformation search set* for short), to be image dependent. Instead, in this construction we aim to find $\mathcal{S}$ to be the smallest set of transformations such that (4) is satisfied. For this purpose, it suffices that the effective search set of images $S^{I'}$ for any image $I' \in \mathbb{G}^I$ covers the corresponding subset $\mathbb{G}^I$ for all I, i.e.

$$S^{I'} := \{g_\Delta(I') : \Delta \in \mathcal{S}\} \supset \mathbb{G}^I.$$

Here we give an explicit construction of $\mathcal{S}$ using the maximal transformation for each subset $\mathbb{G}^I$ that is needed to transform an image of the subset to another. In particular, we define the maximal transformation vector $\Delta^\star \in \mathbb{R}^3$ by the element-wise maximum over all such maximum transformations

$$(\Delta^\star)_j := \max_{I \in \mathcal{I}} \max_{U,U' \in \mathbb{G}^I} |(\Delta)_j| \text{ s.t. } U' = g_\Delta(U)$$

for $j = 1, \ldots, 3$. Although the subsets themselves for each image are not known, using prior knowledge in each application one can usually estimate the largest possible range of transformations $\Delta^\star$ against which robustness is desired or required. For example for images, one could use experiments with humans to determine for which range of angles their reaction time to correctly label each image stays approximately constant. The maximal vector $\Delta^\star$ can now be used to determine the minimal set of transformations $\mathcal{S} = (-\Delta^\star, \Delta^\star)$. A simplified illustration for when $\mathcal{I}$ consists of just one orbit (corresponding for example to one image function and all its rotated variants) can be found in Figure 2.
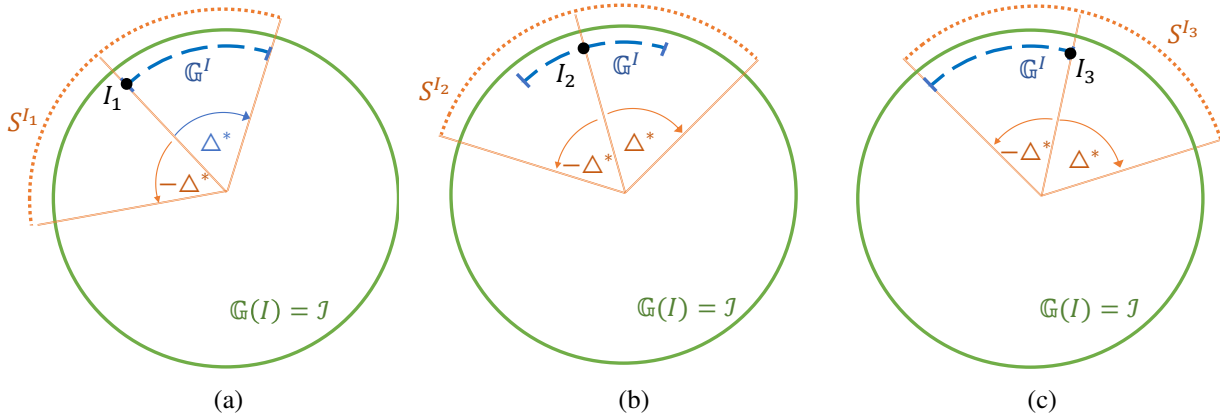

(a) (b) (c)

Figure 2. Illustration of an example where one group orbit $\mathbb{G}(I) =: \mathcal{I}$ is the entire space of images and I is an arbitrary image in the orbit $\mathbb{G}(I)$. We depict one subset of the orbit $\mathbb{G}^I$ and the effective search sets $S^{I'}$ for different instantiations $I' \in \mathbb{G}^I$ defined by the transformation search set $\mathcal{S}$: (a) I_1 on the left boundary of $\mathbb{G}^I$, (b) I_2 in the interior of $\mathbb{G}^I$ and (c) I_3 on the right boundary of $\mathbb{G}^I$. The effective search sets are centered around each instantiation I_j. The necessity of symmetry of the minimal set of transformations $\mathcal{S}$ arises from the requirement to cover $\mathbb{G}^I$ from both boundary points and the maximum transformation vector $\Delta^\star$ that defines $\mathcal{S} = (-\Delta^\star, \Delta^\star)$ is determined by the maximum transformation in $\mathbb{G}^I$ (in blue).

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 sampling an image function $I \in \mathcal{I}$. We use Φ to denote the sampling operator and hence $z = \Phi(I)$. Then the space of

observed finite dimensional images $\mathcal{X}$ is the range space of Φ. 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 so that the quantity $I_z = \Phi^{-1}(z)$ is well-defined. That is, for a finite-dimensional image $z \in \mathcal{X}$, there exists exactly one possible continuous image $I_z \in \mathcal{I}$. As a consequence, if z and a transformed version z' exist in $\mathcal{X}$, then $I_z = I_{z'}$. This is a rather technical assumption that is typically fulfilled in practice. 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 define specific $\mathbb{G}^I$ to be the subsets of $\mathbb{G}(I)$ such that with $z = \Phi^{-1}(I)$, the set $G^z = \{\Phi(I) : I \in \mathbb{G}^{I_z}\}$ corresponds to the support of the marginal distribution $\mathbb{P}$ on $\{\Phi(I) : I \in \mathbb{G}(I_z)\}$. We refer to G^z as transformation sets. By definition of $\mathbb{G}^I$ and bijectivity of Φ, there is an injective mapping from any $z \in \mathcal{X}$ to the set of transformation sets $\mathcal{G}$.

B.6. Proof of Theorem 1

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

We prove the first statement of the theorem by contradiction. Let f^{rob} be the minimizer of $\mathcal{L}_{\text{rob}}(f)$ and let us assume that $f^{\text{rob}} \notin \mathcal{V}$ and in particular that it is constant on all transformation sets except $G^z \in \mathcal{G}$ and the marginal distribution over $\mathcal{G}$ that can be defined as $P(\{G^X = U\}) = P(\{X \in U\})$ for any $U \in \mathcal{G}$, is discrete (for simplicity of presentation) and G^z has non-zero probability.

Let's assume that there is at least one transformation 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 there is a unique mapping Ψ that maps each $x \in \mathcal{X}$ to a unique transformation (see Section B.5), we can lower bound of the robust loss as follows for any $z \in \mathcal{X}$:

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

where the inequality follows from

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

The right hand side is minimized with respect to the set A by choosing $A = \{a^\star\}$ where $a^\star$ is defined as $a^\star = \arg\min_a \int \mathbb{E}_{Y|x} \ell(a, Y) dP_z(x)$ because setting $f^\star(x) = a^\star$ for all $x \in G^z$ and $f^\star(x) = f^{\text{rob}}(x)$ else leads to equality in equation (5) and $f^\star \in \mathcal{F}$ by assumption that $\mathcal{V} \subseteq \mathcal{F}$. Morever, since $\mathbb{P}(\{X \in G^z\}) > 0$ by assumption, choosing $f^\star(x) = a^\star$ for all $x \in G^z$ implies $\mathcal{L}_{\text{rob}}(f^\star) < \mathcal{L}_{\text{rob}}(f^{\text{rob}})$ which contradicts optimality of f^{rob} and thus proves the first statement of the theorem.

For the second statement let us rewrite

$$
\mathcal{L}_{\text{rob}}(f) = \mathcal{L}_{\text{nat}}(f) + [\mathcal{L}_{\text{rob}}(f) - \mathcal{L}_{\text{nat}}(f)]
$$
$$
= \mathbb{E}\, \ell(f(X, Y) + \underbrace{\mathbb{E}\left[\max_{\Delta' \in \mathcal{S}} \ell(f(\mathcal{T}(X, \Delta')), Y) - \ell(f(X), Y)\right]}_{\tilde{R}(f)}
$$

By the first statement we know that the set of invariant functions that minimize the robust loss

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

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

Since $\mathcal{V}(R) \subseteq \mathcal{V}$, 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 because $f^{\min}$ in $\mathcal{V}$ we have $\mathcal{L}_{\text{rob}}(f) = \mathcal{L}_{\text{nat}}(f)$ and 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 $\mathcal{L}_{\text{rob}}(f) = \mathcal{L}_{\text{nat}}(f)$. This concludes the proof of the theorem.

B.7. Proof of Theorem 2

On a high level, similar to the proof of Theorem 1, we can construct a minimizer of the natural loss $f^\star \mathcal{V}$ given the assumption that $Y \perp\!\!\!\perp X|G^z$. Since on $\mathcal{V}$ both losses are equivalent, together with Theorem 1 this shows that the robust minimizer also minimizes the unconstrained natural loss.

Assume $f^{\mathrm{nat}} \notin \mathcal{V}$ minimizes $\mathcal{L}_{\mathrm{nat}}(f)$, and in particular, it is constant on all transformation sets except G^z for some $z \in \mathcal{X}$. Again by existence of a mapping Ψ and by assumption $Y \perp\!\!\!\perp X|G^z$ we can write for any f

$$
\begin{aligned}
\mathcal{L}_{\mathrm{nat}}(f) &= \mathbb{E}_{X,Y}\ell(f(X),Y) \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (6)\\
&= \mathbb{E}\left[\ell(f(X),Y)|X \notin G^z\right]\mathbb{P}(\{X \notin G^z\}) + \mathbb{E}\left[\ell(f(X),Y)|G^z\right]\mathbb{P}(\{X \in G^z\})\\
&= \mathbb{E}\left[\ell(f(X),Y)|X \notin G^z\right]\mathbb{P}(\{X \notin G^z\}) + \mathbb{E}\left[\mathbb{E}_Y[\ell(f(X),Y)]|G^z\right]\mathbb{P}(\{X \in G^z\}).
\end{aligned}
$$

We then obtain

$$
\begin{aligned}
\mathbb{E}\left[\mathbb{E}_Y[\ell(f(X),Y)]|G^z\right] &= \int \mathbb{E}\left[\ell(f(x),Y)|x\right]dP_z(x)\\
&\geq \int \min_{x'\in G^z} \mathbb{E}\left[\ell(f(x'),Y)|x'\right]dP_z(x) = \mathbb{E}\left[\mathbb{E}_Y[\ell(f^\star(X),Y)]|G^z\right] \quad (7)
\end{aligned}
$$

when setting $f^\star(x) = \min_{x\in G^z} \mathbb{E}_Y \ell(f^{\mathrm{nat}}(x),Y)|G^z$ for all $x \in G^z$ and $f^\star(x) = f^{\mathrm{nat}}(x)$ otherwise. Together with equation (6), we thus have that $\mathcal{L}_{\mathrm{nat}}(f^\star) = \mathcal{L}_{\mathrm{nat}}(f^{\mathrm{nat}}) \leq \mathcal{L}_{\mathrm{nat}}(f)$ 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 (7) becomes a strict inequality for $f^{\mathrm{nat}} \notin \mathcal{F}$ and hence we have $\mathcal{L}_{\mathrm{nat}}(f^\star) < \mathcal{L}_{\mathrm{nat}}(f^{\mathrm{nat}})$ 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*.[4]

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. [31, 24], 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. 3 for an example.

C.2. Spatial equivariant networks

We compare the robust prediction accuracies from networks trained with the regularizers with three specialized architectures, designed to be equivariant against spatial transformations and translations: (a) G-ResNet44 (GRN) [7] using p4m convolutional layers (90 degree rotations, translations and mirror reflections) on CIFAR-10; (b) Equivariant Transformer Networks (ETN) [32], a generalization of Polar Transformer Networks (PTN) [11], on SVHN; and (c) Spatial Transformer Networks (STN) [18] on SVHN. A more comprehensive discussion of the literature on equivariant networks can be found

---

[4]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 11).

in Sec. 4. We choose the architectures listed above based on availability of reproducible code and previously reported state-of-the art standard accuracies on SVHN and CIFAR10. We train GRN, STN and ETN using standard augmentation as described in Sec. 2 (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. 3, where the original image is depicted along with a transformed version $\mathcal{T}(\cdot, \Delta)$ with $\Delta \in \mathcal{S}$.
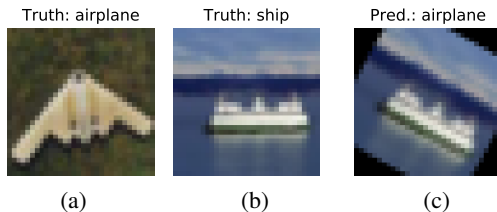

Truth: airplane   Truth: ship   Pred.: airplane

(a)   (b)   (c)

Figure 3. 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 [32], 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+.

LocNet and Classifier   For the classifiers, we take the two models trained on CIFAR-10 and SVHN using standard data augmentation and random rotations from our previous experiments. Since we do not expect the regressors (or LocNets) to be perfect in terms of prediction capability, there will still be some transformation left after the regression stage. Thus, the classifiers should effectively see a smaller range of transformations than without the inclusion of a LocNet and transformer module. The training procedure used to train the classifiers is described in Section 3.

Effect of rendering edges on LocNet   The LocNet 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.

Minimizing loss of information in the prediction transformation process In the spatial transformer module we compare two variants of handling the labels predicted by the LocNet. We can either back-transform the transformed image with the negative 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 LocNet 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".

Observed results For CIFAR-10, this two-stage classifier achieved relatively high grid accuracies. However, the obtained accuracies are still lower than expected, given that the LocNet 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
(rob)	90.95	90.28	64.91	59.68
CIFAR10 (nat)	91.29	90.99	91.29	90.99
(rob)	83.05	84.31	44.88	42.84

C.6. Additional experimental results

In this section we discuss additional experimental results that we collected and and analyzed.

C.7. Stability to selection of regularization parameter λ

Figures 4 and 5 show the test grid and test natural accuracy as a function of the regularization parameter λ. We observe that the regularization methods outperform unregularized methods in terms of grid accuracy in a large range of λ values.

C.8. Additional experimental results

Table 4. Mean accuracies of models trained *without* regularized adversarial training. Standard errors are shown in parentheses.

	std	std*	AT(rob, Wo-10)	AT(mix, Wo-10)
SVHN (nat)	95.48 (0.15)	93.97 (0.09)	96.03 (0.03)	96.56 (0.07)
(rob)	18.85 (1.27)	82.60 (0.23)	90.35 (0.27)	88.83 (0.10)
CIFAR-10 (nat)	92.11 (0.18)	89.93 (0.18)	91.76 (0.23)	93.44 (0.19)
(rob)	9.52 (0.66)	58.29 (0.60)	71.17 (0.26)	68.14 (0.48)
CIFAR-100 (nat)	70.23 (0.18)	66.62 (0.37)	68.79 (0.34)	73.03 (0.13)
(rob)	5.09 (0.25)	28.53 (0.25)	38.21 (0.10)	35.93 (0.24)

Mixed batch experiments In addition to the results reported in the main text, in this section we also report results on more experiments that use the "mixed batch" setting, meaning that the gradient of the loss is taken with respect to both the adversarial and natural examples. This is common practice in the ℓ_p adversarial example literature [20] and we denote this approach by "mix". As can be seen in Table 4, for adversarial training a mixed batch improves natural accuracy at the expense of test grid performance. For the regularization methods, we observe a much small, and not consistent, effect of
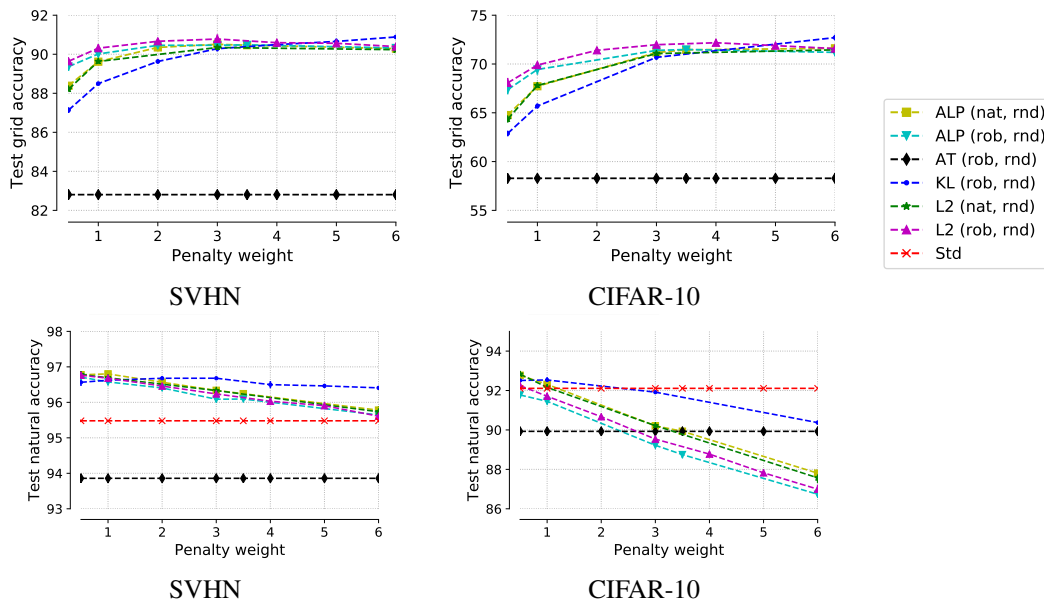
Figure 4. Test grid accuracy (first row) and test natural accuracy (second row) as a function of the regularization parameter λ for the SVHN (first column) and CIFAR-10 (second column) datasets and data augmentation ("rnd"). The test grid accuracy is relatively robust in a large range of λ values while natural test accuracy decreases with larger values of λ.

Figure 5. 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.
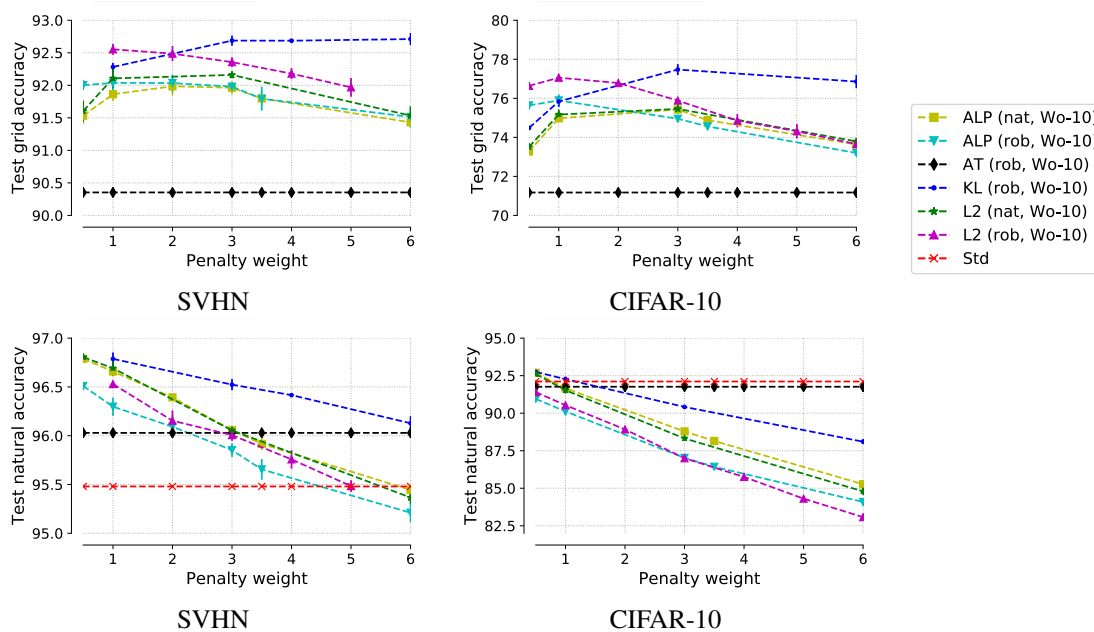
the batch type as can be seen in Table 6. For example, comparing ALP(rob, ·) vs. ALP(mix, ·) shows that the performance differences are mostly not significant.

Table 5. 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)
(rob)	90.69 (0.05)	92.27 (0.09)	92.71 (0.09)	92.42 (0.03)	**92.62** (0.03)
CIFAR-10 (nat)	89.33 (0.16)	90.83 (0.18)	90.41 (0.05)	89.98 (0.21)	89.82 (0.13)
(rob)	73.50 (0.19)	77.34 (0.19)	77.47 (0.28)	**78.93** (0.23)	78.89 (0.07)

Table 6. 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(rob, Wo-10)	ALP(mix, Wo-10)	ALP(rob, Wo-10)	ALP(mix, Wo-20)	ALP(rob, S-PGD)	ALP(mix, S-PGD)
SVHN (nat)	96.05 (0.04)	96.53 (0.03)	96.41 (0.07)	96.3 (0.09)	96.39 (0.04)	96.11 (0.08)	96.30 (0.09)
(rob)	92.16 (0.05)	92.55 (0.08)	92.17 (0.11)	92.04 (0.19)	92.48 (0.05)	92.32 (0.17)	92.42 (0.20)
CIFAR-10 (nat)	88.32 (0.13)	90.53 (0.16)	91.13 (0.13)	90.11 (0.25)	90.67 (0.12)	89.85 (0.27)	89.70 (0.10)
(rob)	75.46 (0.25)	77.06 (0.16)	75.89 (0.23)	75.90 (0.31)	76.72 (0.21)	77.80 (0.17)	77.72 (0.35)
CIFAR-100 (nat)	-	-	68.54 (0.27)	-	68.04 (0.27)	89.82 (0.13)	68.44 (0.39)
(rob)	-	-	49.30 (0.33)	-	49.98 (0.31)	78.89 (0.07)	52.58 (0.20)

Table 7. Mean standard and grid (rob) accuracies of models trained with various forms of regularized adversarial training, using the rnd (equivalent to Wo-1), Wo-10 and Wo-20 defense mechanisms for KL (left) and ALP (right). Standard errors are shown in parentheses.

	KL(nat, Wo-1)	KL(nat, Wo-10)	KL(nat, Wo-20)	ALP(rob, Wo-1)	ALP(rob, Wo-10)	ALP(rob, Wo-20)
CIFAR-10 (nat)	89.34 (0.16)	90.83 (0.18)	89.33 (0.22)	89.47 (0.04)	90.11 (0.25)	90.62 (0.07)
(rob)	73.40 (0.19)	77.34 (0.19)	77.52 (0.16)	73.22 (0.14)	75.90 (0.31)	76.78 (0.15)

Table 8. 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)
(rob)	82.60 (0.23)	90.51 (0.15)	90.69 (0.05)	90.48 (0.16)	90.92 (0.17)	90.48 (0.15)
CIFAR-10 (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)
(rob)	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 9. 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)
(rob)	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)
(rob)	75.43 (0.13)	75.46 (0.25)	76.15 (0.23)	77.34 (0.19)

Weakness of first order attack. Table 10 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.

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

Table 10. Mean accuracies of different models trained with S-PGD defenses and evaluated on the natural test set, against the S-PGD attack and against the grid search attack 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.

Stronger grid search attack To evaluate how much grid accuracy changes with a finer discretization of the perturbation set $\mathcal{S}$, we compare the default grid to a finer one for a subset of experiments, summarized in Table 11. Specifically, "(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. "(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.

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

Table 11. Mean accuracies for different models evaluated against two different grid search attacks. grid-775 represents test grid accuracy using the default grid with 775 transformed examples, grid-7500 shows test grid accuracy on a much finer grid with 7500 transformed examples. Test grid accuracy only decreases slightly for the finer grid and the reduction in accuracy is smaller for ALP than for AT.
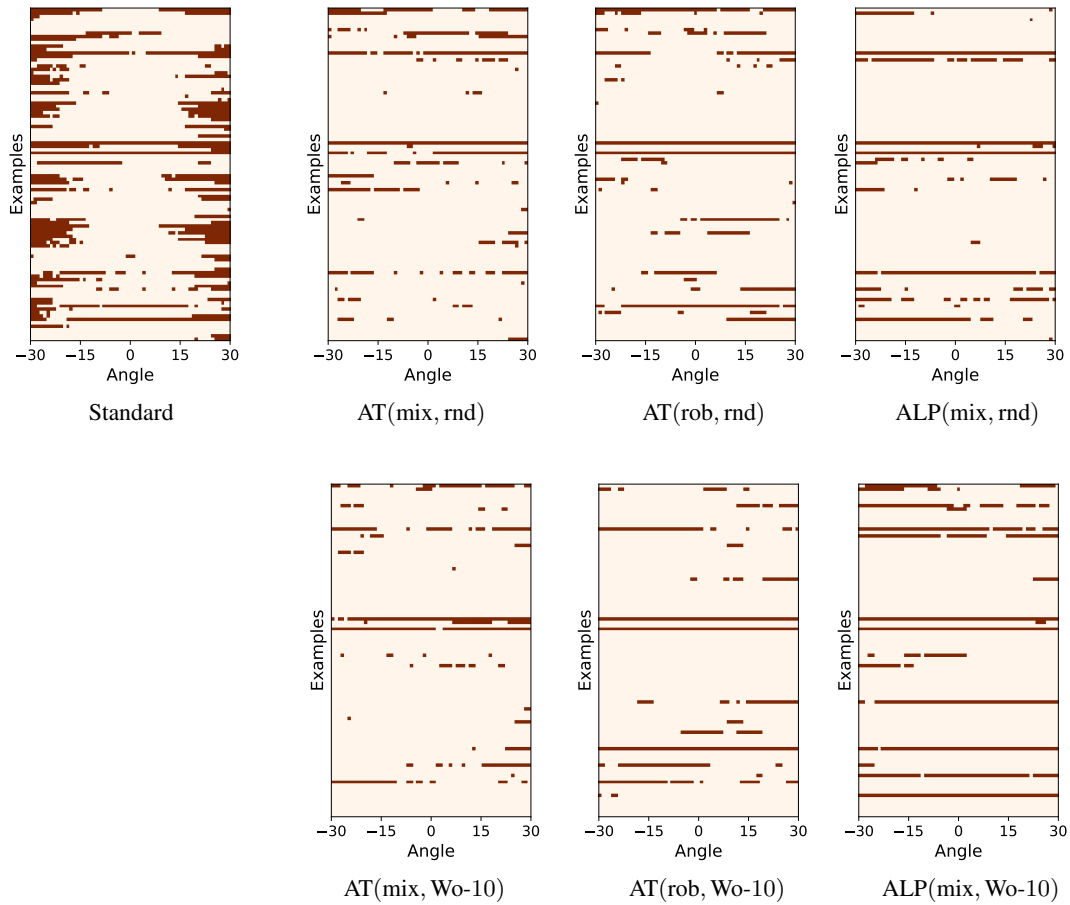
C.9. Regularization effect on range of incorrect angles

Figure 6. For 100 randomly chosen examples from the CIFAR-10 dataset, we show which rotations lead to a misclassification by various models. Each row corresponds to one example and each column to one angle in the interval $[-30°, 30°]$. A dark red square indicates that the corresponding example was misclassified after being rotated by the corresponding angle. The visualization for $\text{AT}(\text{mix}, \text{rnd})$ is more fragmented than for $\text{AT}(\text{rob}, \text{rnd})$ and $\text{ALP}(\text{mix}, \text{rnd})$ and the visualization for $\text{AT}(\text{mix}, \text{Wo-10})$ is more fragmented than for $\text{AT}(\text{rob}, \text{Wo-10})$ and $\text{ALP}(\text{mix}, \text{Wo-10})$.