Supplementary Material for Class-Conditioned Transformation for Enhanced Robust Image Classification

Tsachi Blau[†*] Roy Ganz[†] Chaim Baskin[‡] Michael Elad[†] Alex M. Bronstein[†]

1. Ablation Study

We perform an ablation study, evaluating the influence of different elements on the performance. We use Rebuffi *et al.* [10] AT model WRN28-10 trained on CIFAR10 and threat model $\ell_\infty = 8/255$. We evaluate the performance over clean examples and 4 attacks: $(\ell_\infty, \epsilon = 8/255)$, $(\ell_\infty, \epsilon = 16/255)$, $(\ell_2, \epsilon = 0.5)$, $(\ell_2, \epsilon = 1.0)$. First, we investigate the necessity of the AT model. Next, we check the impact of the distance metric by which we make the prediction. Moreover, we check the influence of the transformation hyper-parameters α, γ, and transformation steps M. Finally, we check different k values for CODIP$_{\text{Top-}k}$.

PAG Property Our method is based on image transformations, and we show the effectiveness of the proposed transformation through extensive evaluation. We claim that the transformation is possible due to the PAG property, which is known to be possessed by AT classifiers [4, 5, 11, 13]. To validate this claim we use our method on WRN28-10 classifier that was trained on clean images. We show that the model enhancement is limited compared to the results achieved by AT models.

Distance Metric Our method operates through two phases: transformation and distance measurement, where the distance is used for the classification decision. CODIP operates through ℓ_2 norm, however, there are other reasonable choices such as LPIPS [16], which measures the perceptual similarity between two images. We demonstrate that ℓ_2 better suit our method.

Hyper-Parameters We investigate the influence of each hyper-parameter, α, γ, M, of CODIP over the model performance. First, we look at different values of α which determines the transformation step size. The proposed transformation is performed via iterative gradient updates, where each step is of size α. Hence, the step size holds an important key. While small steps might better follow the function, the progress is slower since it requires additional steps. On the other hand a large step size, although much faster, might lead to insufficient results. Our results in Tab. 1 support the theory, as a large step size leads to bad results. While small step size, with additional transformation steps, leads to equally good performance.

Next, we examine the influence of γ on the performance. This parameter regulates the transformation distance. Small values allow the transformation to change the image, while large values restrict the transformation to be minimal.

Finally, we examine the number of transformation steps. When the number of steps is small, the transformation can not reach its objective. For a sufficient number of steps, we get good results, even when we have more than the minimal required number of steps.

Top-k We investigate the influence of different k values on CODIP$_{\text{Top-}k}$. For small values, our method performance is getting closer to the AT classifier performances as we rely on its predictions. When we increase k, we rely more on the transformation performance, which leads to better robustness.

[*]tsachiblau@campus.technion.ac.il
[†]Department of Computer Science Technion - Israel Institute of Technology
[‡]School of Electrical and Computer Engineering - Ben-Gurion University of the Negev

Table 1. **Ablation Study** An ablation over (a) The necessity of the AT model, (b) different distance metrics, (c) different hyper-parameters values, and (d) different k values.

AT	Distance	Transformation Steps M	α	γ	Top-k	Clean	Attack L_∞ 8/255	Attack L_∞ 16/255	Attack L_2 0.5	Attack L_2 1.0
✗	–	–	–	–	–	**95.26%**	00.00%	00.00%	00.00%	00.00%
	ℓ_2	30	0.05	300	✗	93.04%	**01.11%**	**01.12%**	**01.24%**	**01.10%**
	LPIPS [16]	30	1.5	400	✗	80.97%	66.49%	33.54%	74.73%	**59.25%**
✓	ℓ_2		0.1	300		**84.86%**	**66.96%**	**35.15%**	**74.84%**	53.32%
✓	ℓ_2	100	0.05	300	✗	**84.86%**	66.91%	35.02%	**74.84%**	**53.50%**
		30	0.05			82.96%	65.40%	33.29%	74.14%	52.74%
			0.1			**84.86%**	66.96%	35.15%	**74.84%**	53.32%
			0.5			82.14%	**67.85%**	**39.11%**	73.32%	52.25%
			1.0			73.78%	63.78%	32.39%	68.52%	47.10%
✓	ℓ_2	30	0.1	10	✗	34.01%	33.84%	29.76%	35.29%	41.65%
				100		81.33%	**67.13%**	**38.47%**	74.02%	**57.05%**
				300		**84.86%**	66.96%	35.15%	**74.84%**	53.32%
				500		84.58%	66.08%	32.78%	73.94%	49.42%
				1000		79.47%	63.63%	29.80%	69.46%	42.41%
✓	ℓ_2	10	0.1	300	✗	70.28%	61.67%	29.66%	69.32%	49.21%
		30				84.86%	66.96%	**35.15%**	74.84%	53.32%
		50				84.87%	66.98%	35.02%	74.76%	53.19%
		100				**84.88%**	**67.01%**	35.13%	**74.85%**	**53.34%**
✓	ℓ_2	30	0.1	300	2	**86.37%**	66.45%	32.70%	73.98%	49.67%
					5	85.79%	**67.12%**	34.72%	**75.14%**	53.30%
					7	85.63%	**67.12%**	**34.97%**	75.12%	**53.55%**
					9	85.21%	66.97%	34.72%	74.91%	53.31%

2. Qualitative Results

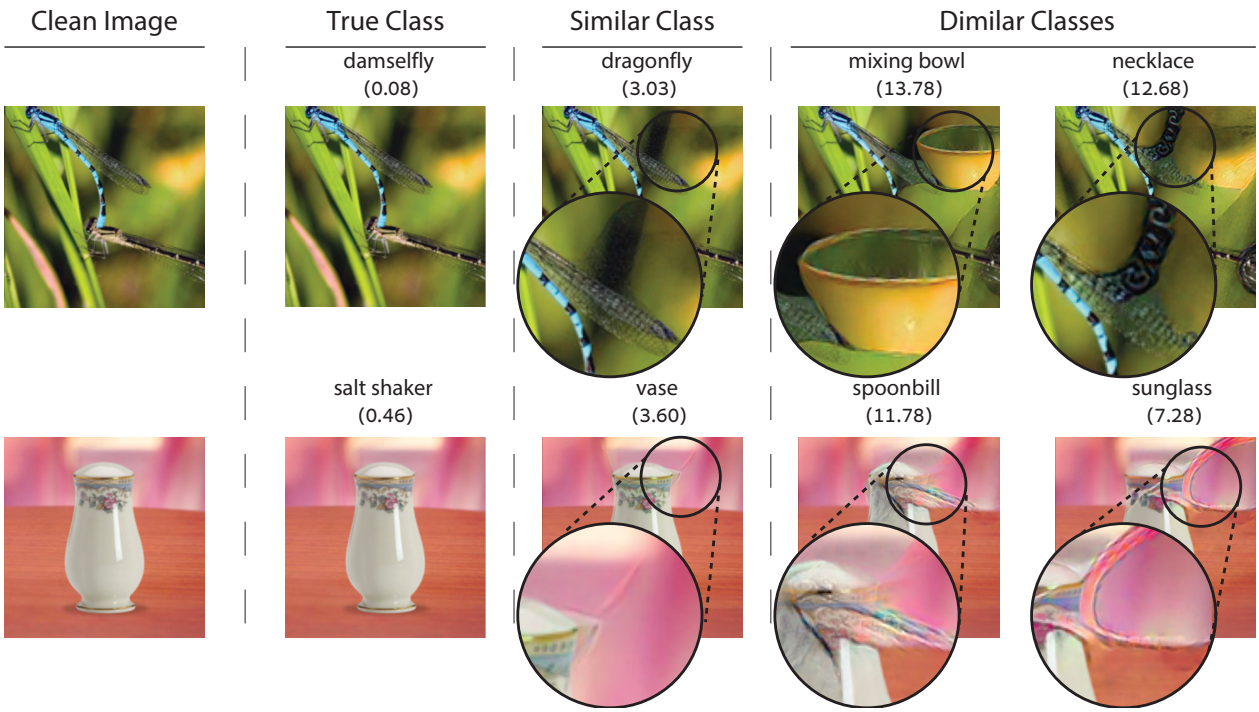

Figure 1. **Class-Conditioned Transformation** Depiction of class-conditioned transformations of clean images from ImageNet towards true class, similar class, and dissimilar class. These transformations performed using CODIP, and the ℓ_2 distance between the clean image and the transformed images is noted below the target class name.

We perform a qualitative experiment in which we compare the transformation's visualization towards different classes. In Fig. 1 we demonstrate that the transformation towards the true class does not change the image appearance much, while transforming to another class does. The transformation to other classes changes the image considerably, especially towards classes that are not semantically similar to the true class, which supports the assumption underlying our method.

3. White-Box Attack

Table 2. **White-Box Attack** An white-box attack evaluation of over CIFAR10 dataset. We compare 'Base', which is the AT classifier, to CODIP$_{\text{Top-}k}$, using PGD attack [9].

AT Method	Trained Threat Model	Architecture	Test-Time Method	Clean	Attack L_∞ 8/255	Attack L_∞ 16/255	Attack L_2 0.5	Attack L_2 1.0
Rebuffi *et al.* [10]	$L_\infty, \epsilon = 8/255$	WRN28-10	Base	**87.33**%	64.40%	33.60%	70.39%	45.78%
			CODIP$_{\text{Top-}k}$	84.38%	**73.13**%	**62.38**%	**79.63**%	**74.64**%

Throughout this work, we followed the common practice [7,8,14], and evaluated the performance with an attack that is not exposed to the whole defense, but to the classifier alone. We perform such an attack since attacking such an iterative defense, with hundreds of steps, is computationally challenging. Yet, there is no empirical evidence that these defenses remain strong when attacking them with a white-box attack.

To this end, we evaluate our method with a white-box attack in Tab. 2, evaluating the robustness of CODIP$_{\text{Top-}k}$ under PGD* [9]. We use the AT method Rebuffi *et al.* [10] and compare the 'Base' defense to CODIP$_{\text{Top-}k}$. As presented, our method improves the base model, by 9% and up to 29% for seen and unseen attacks respectively.

We use the PGD [9] attack that operates through 20 update steps, and we use four threat models, specified In Tab. 3. We evaluate the 'Base' model, which is the AT model without additional test-time defense, and our method CODIP$_{\text{Top-}k}$.

Applying a white-box attack to such an iterative defense is computationally challenging, as we need to keep in memory all of the transformation steps for all of the target classes. This leads to a memory consumption that grows linearly with the number of transformation steps M and the number of target classes N. To overcome this difficulty, even at the cost of decreasing the robust accuracy, we use CODIP$_{\text{Top-}k}$ with $k = 5$ and $M = 20$ transformation steps. Additionally, we set the hyper-parameters $\gamma = 300$ and $\alpha = 0.3$.

*We follow the implementation of https://github.com/MadryLab/robustness

4. Top-k Attack

CODIP$_{\text{Top-}k}$ is our efficient method that selects the Top-k predictions from the classifier. This approach speeds up processing but might be vulnerable to adaptive attacks that exploit this specific mechanism, such as the Top-k attack aimed at excluding the true class from the classifier's Top-k predictions. Typical attacks like PGD [9] focus on reducing the probability of the correct class as much as possible, which can sometimes result in outcomes similar to a Top-k attack. However, achieving this is not their explicit design.

To address this, we introduce two additional adaptive attacks specifically designed to exploit the Top-k vulnerability. The first, Top PGD Out (TPO), performs targeted PGD by iteratively sampling a class ranked outside of the Top-k predictions and attacking towards it. The second, Top PGD In Out (TPIO), extends TPO by also removing the probability of the correct class entirely from the model's predictions, further intensifying the attack's focus on reducing Top-k accuracy. In addition to these attacks, the RCE attack, proposed by Zhang *et al.* [15], is specifically designed to remove the correct class from the Top-k predictions by using normalized cross-entropy loss to update logits in the direction of maximizing the rank distance. This makes RCE a suitable baseline for our evaluation alongside TPO and TPIO.

In Tab. 3, we compare the RCE attack, TPO, and TPIO to the PGD attack, specifically evaluating their effectiveness as Top-k attacks by assessing the mean accuracy of the correct class appearing among the Top-k predictions, rather than general model accuracy. This distinction is crucial, as the reported values reflect Top-k accuracy, not the standard accuracy of the classifier in predicting the most likely class (Top-1 accuracy).

For example, the model by Rebuffi [10] under PGD achieves a 99.05% Top-80 accuracy, meaning that even under a PGD attack, the true class has a 99.05% likelihood of being among the top 80 predictions of the classifier. Our results demonstrate that for k=1, the PGD attack more effectively reduces Top-1 accuracy, achieving a lower Top-1 accuracy compared to the RCE, TPO, and TPIO attacks. For higher k values, RCE perform better, evidenced by slightly lower Top-k accuracy.

We want to emphasize that, despite using standard attacks for all experiments involving CODIP$_{\text{Top-}k}$, rather than specialized Top-k attacks, our results are consistently reliable. We acknowledge the potential risk that a standard attack may not fully exploit the Top-k vulnerability. However, our findings clearly demonstrate that even adaptive attacks such as RCE, TPO, and TPIO, specifically designed to target Top-k vulnerabilities, fail to undermine our defense.

Table 3. **Top-k Attack Analysis** A comparison of Top-k performance under a few attacks, PGD [9], RCE [15] TPO and TPIO, evaluated on CIFAR100 dataset. Each AT model is attacked using two attacks, and the Top-k accuracy is presented for different k values.

Method	Trained Threat Model	Architecture	Attack Threat-Model	Attack Method	Top-k							
					1	5	10	20	30	40	60	80
Rebuffi et al. [10]	$L_\infty, \epsilon = 8/255$	WRN28-10	$L_\infty, \epsilon = 8/255$	PGD [9]	36.18%	64.66%	74.23%	84.44%	89.76%	93.09%	97.40%	99.29%
				RCE [15]	39.12%	64.48%	73.77%	83.52%	89.02%	92.22%	96.95%	99.05%
				TPO	61.71%	84.18%	90.63%	95.43%	97.22%	98.43%	99.46%	99.84%
				TPIO	38.56%	65.10%	74.44%	84.20%	89.61%	92.62%	97.19%	99.07%
Gowal et al. [6]	$L_\infty, \epsilon = 8/255$	WRN70-16	$L_\infty, \epsilon = 8/255$	PGD [9]	40.62%	69.48%	78.05%	86.30%	91.08%	93.86%	97.30%	98.94%
				RCE [15]	43.75%	68.47%	76.73%	84.95%	89.87%	92.81%	96.62%	98.62%
				TPO	61.65%	84.38%	90.64%	95.50%	97.29%	98.53%	99.47%	99.85%
				TPIO	43.73%	68.97%	77.47%	85.58%	90.52%	93.40%	96.99%	98.73%

5. Experimental Details

Table 4. **CODIP Parameters** The parameters used for the main results presented in the paper.

Dataset	Method	Architecture	Trained Threat Model	α	γ
CIFAR10	Madry et al. [9]	RN50	$L_2, \epsilon = 0.5$	1.5	200
	Rebuffi et al. [10]	WRN28-10	$L_2, \epsilon = 0.5$	0.5	400
	Rebuffi et al. [10]	WRN28-10	$L_\infty, \epsilon = 8/255$	0.1	300
	Gowal et al. [6]	WRN70-16	$L_\infty, \epsilon = 8/255$	0.3	300
	Vanila	WRN28-10	-	0.05	300
CIFAR100	Rebuffi et al. [10]	WRN28-10	$L_\infty, \epsilon = 8/255$	0.1	300
	Rebuffi et al. [10]	WRN28-10	$L_\infty, \epsilon = 8/255$	0.1	300
	Gowal et al. [6]	WRN70-16	$L_\infty, \epsilon = 8/255$	0.1	100
	Gowal et al. [6]	WRN70-16	$L_\infty, \epsilon = 8/255$	0.1	100
ImageNet	Madry et al. [9]	RN50	$L_2, \epsilon = 3.0$	6.0	5500
	Salman et al. [12]	WRN50-2	$L_2, \epsilon = 3.0$	6.0	3000
	Madry et al. [9]	RN50	$L_\infty, \epsilon = 4/255$	1.0	6000
	Salman et al. [12]	WRN50-2	$L_\infty, \epsilon = 4/255$	1.0	3000
	Debenedetti et al. [3]	XCiT-S	$L_\infty, \epsilon = 8/255$	0.5	2500
	Debenedetti et al. [3]	XCiT-M	$L_\infty, \epsilon = 8/255$	0.25	3000
Flowers	Debenedetti et al. [3]	XCiT-S	$L_\infty, \epsilon = 8/255$	1.5	500

Table 5. **CODIP Parameters** The parameters used for the results presented in the black-box experiments.

Dataset	Method	Architecture	Trained Threat Model	α	γ
Imagenet	Salman et al. [12]	WRN50-2	$L_2, \epsilon = 3.0$	3.0	10000
	Salman et al. [12]	WRN50-2	$L_\infty, \epsilon = 4/255$	0.1	15000

5.1. Randomized Smoothing

Randomized Smoothing [1] requires that the robust classifier be trained with Gaussian noise augmentations to be effective at test-time. This requirement is documented in [1] and demonstrated by CIFAR-10 results using AT [9] RN50. Despite its limitations, we chose to incorporate this method by selecting a small random noise value, $\sigma = 0.05$, as larger values significantly degrade clean accuracy. Additionally, we attacked this model using AutoAttack [2] (random version), applying 10 augmentations per image to adhere to memory constraints.

6. Explain Top-k

The Top-k version of CODIP sometimes outperforms the full CODIP by focusing on the most likely classes, thereby reducing the influence of less relevant or noisy class transformations. By concentrating on a smaller set of high-confidence predictions, the Top-k version enhances accuracy by avoiding potential noise from less likely classes. For instance, in Fig. 2, we show an image of a tulip (from CIFAR-100). CODIP initially predicts Oak tree followed by Tulip. However, by applying the Top-k filter, Oak tree is excluded, allowing Tulip to be correctly selected. Similarly, for a television image, CODIP might predict Palm tree followed by Television. The Top-k filter excludes Palm tree, leading to the accurate prediction of Television. These examples demonstrate how the Top-k version of CODIP can improve performance by concentrating on the most likely classes and filtering out irrelevant ones.

Figure 2. **CODIP$_{\text{Top-}k}$ Success vs. CODIP Failure.** Two examples from the CIFAR-100 dataset where the Top-k approach succeeds while CODIP fails. The left image shows a correctly classified tulip, while the right image demonstrates another example of correct classification by the Top-k method.

7. Alpha-Controlled Tradeoff for Robustness and Accuracy

The trade-off between clean and robust accuracy is controlled by the step size α, as demonstrated not only on CIFAR-10 but also on larger datasets like ImageNet. In Fig. 3, we illustrate that adjusting α allows for flexible control over this trade-off on ImageNet, further validating our approach. Additionally, while Tab. 1 and Tab. 2 show that clean accuracy may drop, this drop is manageable and adjustable by fine-tuning α. This supports our claim that CODIP can effectively manage the clean-robust accuracy balance, even on more complex datasets with more classes, such as ImageNet, by adjusting the value of α.

Note that Fig. 3 was evaluated on a random subset of 500 examples from the ImageNet validation set, as it is intended for exemplification purposes.

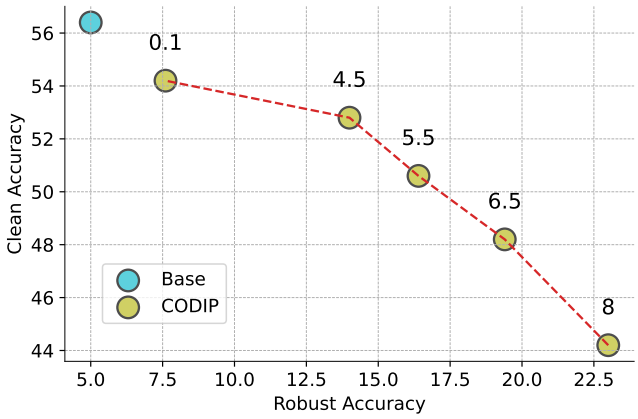

Figure 3. **Imagenet Clean-Robust Accuracy Trade-off** A demonstration of our proposed controlled clean-robust accuracy tradeoff. The tradeoff is controlled by adapting the step size value α, specified beside each of CODIP workpoints. The used test-time methods 'Base' which is the base AT model.

8. γ Effect on Clean-Robust Accuracy Trade-off

As demonstrated in Fig. 3, the hyperparameter γ plays a critical role in controlling the clean-robust accuracy trade-off across different values of α. Both α and γ contribute to restricting the transformation in order to maintain the accuracy of the model. While α primarily governs the magnitude of the perturbation, γ regularizes the transformation by ensuring that it remains subtle enough to preserve the natural characteristics of the image while successfully changing its class, even in the presence of small-norm perturbations.

When $\gamma = 0$, the transformation is unregularized, leading to a situation where both clean and robust accuracy suffer significantly. Without the restriction provided by γ, the transformation can deviate excessively from the input image, negatively impacting both clean accuracy and robustness. This issue is illustrated in Fig. 4, where we compare transformations produced by CODIP and targeted PGD. The image transformed by CODIP (right) maintains a closer resemblance to the original input image, while the image generated by targeted PGD (left) appears unnatural and distorted. The ℓ_2 distances further highlight this: CODIP achieves a much lower distance (11.22) compared to targeted PGD (42.97). In addition, in Fig. 3, we show that both robust and clean accuracy metrics are substantially worse when $\gamma = 0$, emphasizing that an unregularized transformation fails to achieve the desired balance between robustness and preserving the integrity of the original input.

To achieve the optimal balance between clean and robust accuracy, it is essential to carefully tune γ. In Fig. 3, we present three graphs that illustrate the impact of different γ values across various α settings. In each experiment, increasing γ initially improves clean accuracy. For larger α values, increasing γ significantly enhances robustness, though after a certain threshold, clean accuracy may begin to slightly decrease. For smaller α, robust accuracy increases slightly before eventually decreasing. These findings demonstrate that while careful tuning of γ is necessary, it remains a critical parameter for managing the clean-robust trade-off, as it controls how closely the transformation adheres to the original image, especially under varying perturbation magnitudes. Specifically, we show its pivotal role in controlling clean accuracy

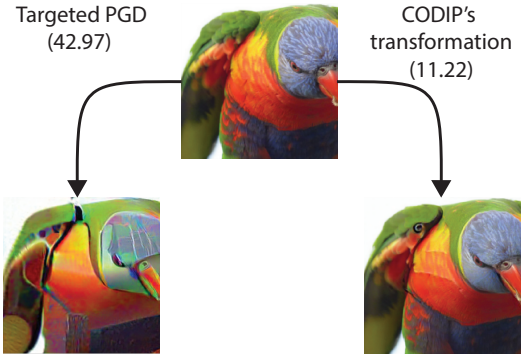

Figure 4. **CODIP vs. Targeted PGD** A comparison between CODIP and targeted PGD class conditioned transformations. An image of *lorikeet* is transformed into a *toucan* using CODIP and targeted PGD. A ℓ_2 distance between the clean image and the transformed ones is stated beneath the attack name.

References

[1] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 6

[2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 6

[3] Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 225–253. IEEE, 2023. 6

[4] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019. 1

[5] Christian Etmann, Sebastian Lunz, Peter Maass, and Carola-Bibiane Schönlieb. On the connection between adversarial robustness and saliency map interpretability. *arXiv preprint arXiv:1905.04172*, 2019. 1

[6] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 5, 6

[7] Mitch Hill, Jonathan Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. *arXiv preprint arXiv:2005.13525*, 2020. 4

[8] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020. 4

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 4, 5, 6

[10] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. 1, 4, 5, 6

[11] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1

[12] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020. 6

[13] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1

[14] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, pages 12062–12072. PMLR, 2021. 4

[15] Chaoning Zhang, Philipp Benz, Adil Karjauv, Jae Won Cho, Kang Zhang, and In So Kweon. Investigating top-k white-box and transferable black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15085–15094, 2022. 5

[16] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1, 2