EVALUATING MODEL ROBUSTNESS AGAINST UNFORESEEN ADVERSARIAL ATTACKS

Anonymous authors

003 004

006

008 009

010 011

012

013

014

015

016

017

018

019

021

023

Paper under double-blind review

ABSTRACT

When considering real-world adversarial settings, defenders are unlikely to have access to the full range of deployment-time adversaries, and adversaries are likely to use realistic adversarial distortions that will not be limited to small L_p -constrained perturbations. To narrow in on this discrepancy between research and reality we introduce ImageNet-UA, a new benchmark for evaluating model robustness against a wide range of *unforeseen adversaries*. We make use of our benchmark to identify holes in current popular adversarial defense techniques, highlighting a rich space of techniques which can improve unforeseen robustness. We hope the greater variety and realism of ImageNet-UA will make it a useful tool for those working on real-world worst-case robustness, enabling development of more robust defenses which can generalize beyond attacks seen during training.

1 INTRODUCTION

025 Neural networks perform well on a variety of tasks, yet can be consistently fooled by minor adversarial 026 distortions (Szegedy et al., 2013; Goodfellow et al., 2014b). This has led to an extensive area of 027 research around the " L_p -bounded adversary" that adds imperceptible distortions to model inputs 028 to cause misclassification. However, this classic threat model may fail to fully capture many real-029 world concerns regarding worst-case robustness (Gilmer et al., 2018). Firstly, real-world worst-case distributions are likely to be varied, and are unlikely to be constrained to the L_p ball. Secondly, developers will not have access to the worst-case inputs to which their systems will be exposed to. 031 For example, online advertisers use perturbed pixels in ads to defeat ad blockers trained only on the previous generation of ads in an ever-escalating arms race (Tramèr et al., 2018). Furthermore, 033 although research has shown that adversarial training can lead to overfitting, wherein robustness 034 against one particular adversary does not generalize (Dai et al., 2022; Yu et al., 2021; Stutz et al., 2020; Tramer & Boneh, 2019), the existing literature is still focuses on defenses that train against the test-time attacks. Although such distribution shifts have been studied in the average-case common 037 corruption setting (Hendrycks & Dietterich, 2018), when considering worst-case inputs, the research 038 community is lacking a unified benchmark for testing how defences generalise.

We address the limitations of current adversarial robustness evaluations by providing a repository of nineteen gradient-based attacks, eight of which are used to create ImageNet-UA—a benchmark for evaluating the *unforeseen robustness* of models on the popular ImageNet dataset (Deng et al., 2009). Defenses achieving high Unforeseen Adversarial Accuracy (UA2) on ImageNet-UA generalize to a diverse set of adversaries not seen at train time, demonstrating robustness to a much more realistic threat model than the L_p adversaries which are a focus of the literature.

045 Our results show that unforeseen robustness is distinct from existing robustness metrics, further 046 highlighting the need for a new measure which better captures the generalization of defense methods. 047 We use ImageNet-UA to reveal that models with high L_{∞} attack robustness (the most ubiquitous 048 measure of robustness in the literature) do not generalize well to new attacks, recommending L_2 as a stronger baseline. We further find that L_p training can be improved on by alternative training processes, and suggest that the community focuses on training methods with better generalization 051 behavior. Interestingly, unlike in the L_p case, we find that progress on CV benchmarks has at least partially tracked unforeseen robustness. We hope that the community can build on these results, 052 so that ImageNet-UA can provide an improved progress measure for defenses aiming to achieve real-world worst-case robustness. To summarize, we make the following contributions:





• We carry out in-depth analysis of the behaviour of different defence techniques on ImageNet-UA. We demonstrate that UA2 is distinct from existing robustness metrics in the literature, and point towards several promising directions for improving the generalisation of adversarial defences.

2 RELATED WORK

Evaluating Adversarial Robustness. Adversarial robustness is notoriously difficult to evaluate correctly (Papernot et al., 2017; Athalye et al., 2018). To this end, Carlini et al. (2019) provide extensive guidance for sound adversarial robustness evaluation. Our ImageNet-UA benchmark incorporates several of their recommendations, such as measuring attack success rates across several magnitudes of distortion and using a broader threat model with diverse differentiable attacks. Existing popular measures of adversarial robustness (Croce & Hein, 2020; Moosavi-Dezfooli et al., 2015; Weng et al., 2018) mainly present novel optimisation techniques for optimizing over an L_p -ball, limiting their applicability for modeling robustness to new deployment-time adversaries.

Non- L_p **Attacks.** Attacks often use hard-to-bound generative models (Song et al., 2018; Qiu et al., 2019), or make use of expensive brute-force search techniques Engstrom et al. (2017). We focus on attacks which are fast by virtue of differentiability, portable across datasets and independent of auxiliary generative models. Previous works presenting suitable attacks include Laidlaw & Feizi (2019); Shamsabadi et al. (2021); Zhao et al. (2019), who all transform the underlying color space of



Figure 2: Progression of an attack. As we optimize our differentiable corruptions, model perfor-126 mance decreases, while leaving the image semantics unchanged. Unoptimized versions of our attacks have a moderate impact on classifier performance, similar to common corruptions (Hendrycks & Dietterich, 2019), while optimized versions are able to cause large drops in model accuracy. 128

127

131 an image and Xiao et al. (2018) who differentiably warp images— a technique which we adapt to 132 create our own Elastic adversary.

133 Unforeseen and Multi-attack Robustness. There exist defense methods which seek to generalize 134 across an adversarial train-test gap (Dai et al., 2022; Laidlaw et al., 2020; Lin et al., 2020). Yet, 135 comparison between these methods is challenging due to the lack of a standardized benchmark 136 and an insufficient range of adversaries to test against. We fill this gap by implementing a unified 137 benchmark for testing unforeseen robustness. The more developed field of multi-attack robustness 138 (Tramer & Boneh, 2019) aims to create models which are robust to a range of attacks, but works 139 generally focus on a union of L_p adversaries (Maini et al., 2020; Madaan et al., 2021a; Croce & Hein, 2022) and do not enforce that test time adversaries have to differ from those used during training. 140

141 **Common corruptions** Several of our attacks (Pixel, Snow, JPEG and Fog) were inspired by existing 142 common corruptions (Hendrycks & Dietterich, 2018). We fundamentally change the generation 143 methods to make these corruptions differentiable, allowing us to focus on worst-case robustness 144 instead of the average-case robustness (see Section 4.1 for a comparison between the two benchmarks).

145 146 147

148

149

150

3 MEASURING UNFORESEEN ROBUSTNESS

To evaluate the unforeseen robustness of models, we introduce a new benchmark ImageNet-UA, and corresponding metric UA2 (Unforeseen Adversarial Accuracy). ImageNet-UA consists of our eight core adversarial attacks pictured in Figure 5, which have been selected for efficacy and computational efficiency.

- 151 152 153 154
- 3.1 THE UNFORESEEN ROBUSTNESS THREAT MODEL

155 The unforeseen robustness threat model has three central features:

156 Test-time distribution shift. As illustrated in Figure 5, our threat model requires that defenses do not 157 make use of our test-time adversaries. This is directly analogous to the situation faced by developers, 158 who are unable to anticipate which worst-case inputs which may occur at deployment time. 159

A diverse population of adversaries. Defence techniques should generalise to a range of possible 160 deployment-time adversaries (e.g. they should not be effective only against L_p -based adversaries). 161 Testing generalisation on a diverse range of adversaries is therefore an important part of evaluation.



Figure 3: An illustrative example of the generating process for one of our attacks. As demonstrated by this illustration of our Wood attack, all of our attacks function by performing PGD optimization on a set of latent variables. In the case of the Wood attack, these latent variables are inputs to concentric sine waves $(F(x, y) = \sin(\sqrt{x^2 + y^2}))$ which are overlaid on the image. See Appendix C for a more detailed explanation. We design effective attacks which are fast, easy to optimize, precisely bound, preserve image semantics, are portable across datasets and have variable intensity through the ε parameter.

195

196 197

201 202

205

209

White-box access to models. To ensure the strength of our adversaries (Carlini et al., 2019), and to avoid the usage of computationally expensive black-box optimization techniques, adversaries have full white-box access to victim models.

Unlike the more classical L_p threat models (Szegedy et al., 2013; Goodfellow et al., 2014a), we do not require that distortions are imperceptible (although they must preserve semantics, see Appendix K for a human study). This is because in many realistic settings, attackers are unlikely to face these constraints.

3.2 GENERATING ADVERSARIAL EXAMPLES

To test the performance of defence techniques, we must create a range of adversaries (see Section 3.3). Each of these adversaries are defined using a differentiable function A, which generates an adversarial input x_{adv} from an input image x and some latent variables δ :

$$x_{\text{adv}} = A(x, \delta). \tag{1}$$

To control the strength of our adversary, we introduce an L_p constraint to the variables δ , by bounding the size using some constraints ε_A :

$$\|\delta\|_p \le \varepsilon_A$$

As is typical in the literature (Madry et al., 2017b), we use our dataset loss function \mathcal{L} to re-frame the finding of adversarial examples in our perturbation set as a continuous optimisation problem, seeking δ_{adv} which solves:

$$\delta_{\text{adv}} = \underset{\delta: \|\delta\|_{P} \le \varepsilon}{\operatorname{argmin}} \{ \mathcal{L}(f(A(x, \delta)), y) \},$$
(2)

To solve this, we then use the popular method of Projected Gradient Descent (PGD) (Madry et al., 2017b) to find an approximate solution to Equation (2).

Using this formulation helps us ensure that all of our attacks are independent of auxiliary generative
models, add minimal overhead when compared to the popular PGD adversary (see Appendix E),
are usable in a dataset-agnostic "plug-and-play" manner, can be used with existing optimization
algorithms (see Figure 4a for behavior of attacks under optimization), come with a natural way of

Table 1: L_p robustness is disctinct from unforeseen robustness. We highlight some of the models which achieve high UA2, while still being susceptible to L_p attacks. Models below the dividing line are adversarially trained, with norm constraints in parentheses.

Model	$L_{\infty} \ (\varepsilon = 4/255)$	UA2
Dinov2 Vit-large	27.7	27.2
Convnext-V2-large IN-1k+22K	0.0	19.2
Swin-Large ImageNet1K	0.0	16.2
ConvNext-Base L_{∞} , ($\varepsilon = 8/255$)	58.0	22.3
Resnet-50, L_{∞} ($\varepsilon = 8/255$)	38.9	10
Resnet-50 L_2 , ($\varepsilon = 5$)	34.1	13.9

varying intensity through adjusting ε parameter (see Figure 4b for behavior under varying ε), and have precisely defined perturbation sets. As discussed in Section 2, this is not the case for most existing attacks in the literature, prompting us to design our set of new attacks.

234 235

232

233

236

3.3 EIGHT CORE UNFORESEEN ATTACKS

237 238

To ensure that we have high quality and diversity of tasks, we design nineteen novel attacks (see
Appendix B for a full list), and select a core eight for their computational efficiency, effectiveness,
variety and preservation of semantics (see Appendix D for further discussion, and Appendix K for a
human study on semantic preservation). We use these attacks to create ImageNet-UA.

Importantly, we do not claim that these attacks provide an exhaustive taxonomy over adversaries.
They simply represent a specific set of diverse held-out adversaries— mirroring methodologies that
have been popular in other studies of distribution shift (Hendrycks & Dietterich, 2018; Hendrycks
et al., 2021; Wang et al., 2019). We briefly describe our core attacks below:

Wood. The wood attack is described in Figure 3 and Appendix C.

Glitch. Glitch simulates a common behavior in corrupted images of colored fuzziness. Glitch greys
 out the image, splitting it into horizontal bars, before independently shifting color channels within
 each of these bars.

JPEG. The JPEG compression algorithm functions by encoding small image patches using the discrete cosine transform, and then quantizing the results. The attack functions by optimizing L_{∞} -constrained perturbations within the JPEG-encoded space of compressed images and then reverse-transforming to obtain the image in pixel space, using ideas from Shin & Song (2017) to make this differentiable.

Gabor. Gabor spatially occludes the image with visually diverse Gabor noise (Lagae et al., 2009),
 optimizing the underlying sparse tensor which the Gabor kernels are applied to.

Kaleidoscope. Kaleidoscope overlays randomly colored polygons onto the image, and then optimizes both the homogeneous color of the inside of the shape, and the darkness/lightness of the individual pixels on the shape's border, up to an L_{∞} constraint.

Pixel. Pixel modifies an image so it appears to be of lower quality, by first splitting the image into $m \times m$ "pixels" and then and averaging the image color within each block. The optimization variables δ then control the level of pixelation, on a per-block bases.

Elastic. Our only non-novel attack. Elastic is adapted from (Xiao et al., 2018), functioning by which warping the image by distortions x' = Flow(x, V), where $V : \{1, ..., 224\}^2 \rightarrow \mathcal{R}^2$ is a vector field on pixel space, and Flow sets the value of pixel (i, j) to the bilinearly interpolated original value at (i, j) + V(i, j). To make the attack suitable for high-resolution images, we modify the original attack by passing a Gaussian kernel over V.



Figure 4: Attack effectiveness increases with optimization pressure and distortion budget. We 285 average performance against our core attacks across all our benchmarked models, demonstrating 286 that our attacks respond to increased optimization pressure (Figure 4a). We further demonstrate the importance of the gradient-based nature by comparing random grid search to our gradient-288 based method in Appendix M. Furthermore, we demonstrate the ability for our attack stength to be customisable by showing that increasing distortion budget reduces model performance (Figure 4b).

287

289

292 Snow. Snow functions by optimising the intensity of individually snowflakes within an image, 293 which are created by passing a convolutional filter over a sparsely populated tensor, and then optimising the non-zero entries in this tensor.

295 Due to the time-consuming nature of designing new non- L_p adversarial attacks, most previous works 296 only present and analyse a single adversarial attack (Laidlaw & Feizi, 2019; Shamsabadi et al., 2021; 297 Zhao et al., 2019; Xiao et al., 2018). Hence, we believe that the additional attacks are a valuable 298 resource for the community to build on, and release the full set of 19 publicly. We are particularly 299 excited for these attacks as a basis of future evaluations of potential over-fitting to our original set of attacks, mirroring what was done by Mintun et al. (2021) 300

301 302

303

3.4 ImageNet-UA: A NEW BENCHMARK FOR UNFORESEEN ROBUSTNESS

We introduce ImageNet-UA, a benchmark for evaluating the unforeseen robustness of image classi-304 fiers on the popular ImageNet dataset (Deng et al., 2009). We also develop a CIFAR-10 equivalent, 305 which we call CIFAR-10-UA, for computationally efficient evaluation of defense strategies and attack 306 methods. We performed extensive sweeps to find the most effective hyperparameters for all of our 307 attacks, the results of which are in Appendix A. 308

We quantify the unforeseen robustness achieved by a defense with Unforeseen Adversarial Accuracy 309 (UA2), which measures the robustness of a given classifier f across a diverse range of attacks 310 that are not seen at training time. We model the deployment-time population of adversaries as a 311 uniform distribution over some finite set of adversaries A. Our accuracy against such a population of 312 adversaries is the average accuracy across each of the individual attacks: 313

314 315

316 317

$$\mathsf{UA2} := \frac{1}{|\mathbf{A}|} \sum_{A \in \mathbf{A}} \operatorname{Acc}(A, \epsilon_A, f)$$

where $Acc(A, \varepsilon_a, f)$ denotes the adversarial accuracy of classifier f against attack A at distortion 318 level ϵ_A . We select the population of adversaries to be the eight core adversaries from Section 3.3, 319 setting **A** = {JPEG, Elastic, Wood, Glitch, Kaleidoscope, Pixel, Snow, Gabor}. 320

321 We further divide our benchmark by picking three different distortion levels for each attack, leading to three different measures of unforeseen robustness: $UA2_{low}$, $UA2_{med}$ and $UA2_{high}$ (see Appendix A 322 for specific ε values used within this work), and we focus on focus on UA2_{med} for all of our reports, 323 referring to this distortion level as simply UA2. As distortion levels increase, model performance

Table 2: ImageNet-UA **baselines** Here, we show some of the most robust models on ImageNet-UA, as well as baseline ResNet-50 models to compare between. We see a variety of techniques achieving high levels of robustness, demonstrating a rich space of possible interventions. The L_{∞} column tracks robustness against a PGD L_{∞} adversary with $\varepsilon = 4/255$. Numbers denote percentages.

Model	Clean Acc.	L_{∞}	UA2	JPEG	Elastic	Wood	Glitch	Kal.	Pixel	Snow	Gabor
DINOv2 ViT-large Patch14	86.1	15.3	27.7	14.3	42.6	39.7	17.7	46.2	17.2	14.2	29.9
ConvNeXt-V2-large IN-1K+22K	87.3	0.0	19.2	0.0	39.1	34.4	21.4	16.1	15.5	4.0	23.1
ConvNeXt-V2-huge IN-1K	86.3	0.0	17.7	0.0	42.5	21.2	23.8	24.3	6.6	0.7	22.2
ConvNeXt-base, L_{∞} (4/255)	76.1	58.0	22.3	39.0	23.8	47.9	12.9	2.5	9.7	30.2	12.8
ViT-base Patch16, L_{∞} (4/255)	76.8	57.1	25.8	52.6	26.3	47.2	13.8	8.1	11.9	27.1	19.5
Swin-base IN-1K	85.3	0.0	15.2	0.0	31.4	24.6	16.2	6.0	6.9	4.3	32.0
ResNet-50	76.1	0.0	1.6	0.0	4.4	6.3	0.4	0.0	0.3	0.1	0.9
ResNet-50 + CutMix	78.6	0.5	6.1	0.2	17.9	15.5	2.5	0.1	6.7	3.0	2.7
ResNet-50, L_{∞} (8/255)	54.5	38.9	10.0	6.9	11.8	23.9	14.4	0.7	5.2	15.6	1.2
ResNet-50, L_2 (5)	56.1	34.1	13.9	39.7	11.9	19.4	12.2	0.3	9.7	15.4	2.5

Table 3: L_p training. We train a range of ResNet-50 models against L_p adversaries on ImageNet-UA

Training	Train ε	Clean Acc.	UA2
Standard	-	76.1	1.6
L_2	$\begin{array}{c} 1\\ 3\\ 5\end{array}$	69.1 62.8 56.1	6.4 12.2 13.9
L_{∞}	$2/255 \\ 4/255 \\ 8/255$	69.1 63.9 54.5	6.4 7.9 10.0

Table 4: L_p training on generated data. We see the effect of training when training WRN-28-10 networks on CIFAR-10-50M, a 1000x larger diffusion-model generated version of CIFAR-10 (Wang et al., 2023)

Dataset	Training	Clean Acc.	UA2
CIFAR-10	$L_2, \varepsilon = 1 L_\infty, \varepsilon = 8/255$	82.3 86.1	45.8 41.5
CIFAR-10-50M	$L_2, \varepsilon = 0.5 \\ L_\infty, \varepsilon = 4/255$	95.2 92.4	51.2 51.5

decreases (Figure 4b). We perform a human study (Appendix K) to ensure UA2_{med} preserves image semantics.

4 EXPERIMENTS

328

340

341

342

343

345

353

354 355 356

357

360

361 362

364

In this section, we evaluate a range of models on our standardized benchmarks ImageNet-UA and CIFAR-10-UA. We aim to present a set of directions for future work, by comparing a wide range of methods. We also hope to explore how the problem of unforeseen robustness differs from existing robustness metrics.

4.1 How do existing robustness measures relate to unforeseen robustness?

We find differences between existing metrics and UA2, suggesting that the setting of unforeseen adversarial robustness may require new methods to obtain strong performance.

367 UA2 is distinct from existing measures of distribution shift. We compare UA2 to several standard 368 distribution-shift benchmarks—ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-R Hendrycks 369 et al. (2021) and ImageNet-Sketch (Wang et al., 2019). As shown in Table 5 and Appendix I, performance on these benchmark is similar to performance on non-optimized versions of our attacks. By 370 contrast, the optimized versions of our attacks are far more challenging and have distinct properties. 371 For example, while adversarial training does not improve performance on ImageNet-C, it does im-372 prove performance on ImageNet-UA. This highlights that UA2 is a measure of worst case robustness, 373 similar to L_p robustness, and distinct from other distribution shift benchmarks in the literature. 374

375 L_p robustness is correlated, but distinct from, unforeseen robustness. As shown in Appendix L, 376 unforeseen robustness is correlated with L_p robustness. Our attacks also show similar properties to 377 L_p counterparts, such as the ability for black-box transfer (Appendix N). However, many models show susceptibility to L_p adversaries while still performing well on UA2 (Table 1), and a range

Table 5: Common corruptions and UA2 We compare ImageNet-C to both non-optimized and
 optimized versions of our attacks. We find that ImageNet-C behaves similarly to our non-optimized
 attacks, while our optimised attacks are far more challenging.

Model	UA2 (non-optimized) \uparrow	mCE \downarrow	UA2 \uparrow
Resnet 50	55.2	76.7	1.6
Resnet50 + AugMix	59.1	65.7	3.5
Resnet50 + DeepAug	60.2	61.1	3.0
Resnet50 + Mixup	59.9	69.2	4.8
Resnet50 + L_2 , ($\varepsilon = 5$)	43.2	89.0	13.9
Resnet50 + L_{∞} , ($\varepsilon = 8/255$)	40.6	85.1	10

389 390

382

384

386 387 388

391 392

393

394

of strategies beat L_p training baselines Section 4.2. We conclude that UA2 is distinct from L_p robustness, and present UA2 as an improved progress measure when working towards real-world worst-case robustness.

 L_2 -based adversarial training outperforms L_∞ -based adversarial training. We see that L_p adversarial training increases the unforeseen robustness of tested models, with L_2 adversarial training providing the largest increase in UA2 over standard training (1.6% \rightarrow 13.9%), beating models which are trained against L_∞ adversaries (1.6% \rightarrow 10.0%). We present L_2 trained models as a strong baseline for unforeseen robustness, noting that the discrepancy between L_∞ and L_2 training is particularly relevant as L_∞ robustness is the most ubiquitous measure of adversarial robustness.

400 401 402

4.2 How can we improve Unforeseen Robustness?

We find several promising directions that improve over standard L_p training. These include standard multi-attack robustness methods and also novel methods that combine insights from different areas of robustness research.

Combining image augmentations and L_{∞} training. One simple approach to improving robustness to unforeseen adversaries would be to combine insights from distribution shift robustness and adversarial robustness research. To explore this avenue, we experiment with a combination of PixMix and L_{∞} adversarial training, applying adversarial perturbations to the augmented images from PixMix. We show the results of this experiment in Table 6.

Surprisingly, we find that this simple approach can be highly effective. Namely, we find that combining PixMix and adversarial training gives a UA2 of 45.5 percent, compared to 37.3 with adversarial training alone. This novel training strategy beats strong baselines by combining two distinct robustness techniques. The surprising effectiveness of this simple method highlights how unforeseen robustness may foster the development of new methods.

416 Multi-attack robustness. To evaluate how ex-417 isting work on robustness to a union of L_p 418 balls may improve unforeseen robustness, we 419 use CIFAR-10-UA to evaluate a strong multi-420 attack robustness baseline by (Madaan et al., 421 2021b), which trains a Meta Noise Generator 422 (MNG) that learns the optimal training pertur-423 bations to achieve robustness to a union of L_p adversaries. For WRN-28-10 models on 424 CIFAR-10-UA, we see a large increase in un-425 foreseen robustness compared to the best L_p 426 baseline $(21.4\% \rightarrow 51.1\%)$, full results in Ap-427 pendix J), leaving scaling of such methods to 428 full ImageNet-UA for future work. 429

- 430 Bounding perturbations with perceptual dis
 - tance. We evaluate the UA2 of models trained

Table 6: **PixMix and** L_p **training.** We compare UA2 on CIFAR-10 of models trained with PixMix and adversarial training. Combining PixMix and adversarial training improves UA2, demonstrating the potential for novel methods to improve UA2. All numbers denote percentages, and L_{∞} training was performed with the TRADES algorithm.

Training Strategy	Train ε	Clean Acc.	UA2
PixMix	-	95.1	15.00
L_{∞}	4/255	89.3	37.3
L_{∞} + PixMix	4/255	91.4	45.1
$\begin{array}{c} L_{\infty} \\ L_{\infty} + \operatorname{PixMix} \end{array}$	8/255	84.3	41.4
	8/255	87.1	47.4

with Perceptual Adversarial Training (PAT) (Laidlaw et al., 2020). PAT functions by training a

Table 7: **Effects of data augmentation on** UA2. We evaluate the UA2 of a range of data-augmented ResNet50 models.

Clean Acc.

76.1

79.1

78.6

75.8

Table 8: Effects of pretraining and regular-ization on UA2.

Model	Clean Acc.	UA2
ConvNeXt-V2-28.6M	83.0	9.8
ConvNeXt-V1-28M	82.1	5.1
ConvNeXt-V2-89M	84.9	14.9
ConvNeXt-V1-89M	83.8	9.7
ConvNeXt-V2-198M	85.8	19.1
ConvNeXt-V1-198M	84.3	10.6

model against an adversary bounded by an estimate of the human perceptual distance, computing the estimate by using the hidden states of an image classifier. For computational reasons we train and evaluate ResNet-50s on a 100-image subset of ImageNet-UA, where this technique outperforms the best L_p trained baselines (22.6 \rightarrow 26.2, full results in Appendix J).

UA2

1.0

6.0

6.0

1.8

Regularizing high-level features. We evaluate Variational Regularization (VR) (Dai et al., 2022), which adds a penalty term to the loss function for variance in higher level features. We find that the largest gains in unforeseen robustness come from combining VR with PAT, improving over standard PAT ($26.2 \rightarrow 29.5$, on a 100 class subset of ImageNet-UA, full results in Appendix J).

452 453 454

432

433

434

435

436 437

438

439

440

441

446

447

448 449

450

451

Training

Standard

Moex

CutMix

Deepaugment + Augmix

455 456 4.3 HOW HAS PROGRESS ON CV BENCHMARKS TRACKED UNFORESEEN ROBUSTNESS?

457 **Computer vision progress has partially tracked unforeseen robustness.** Comparing the UA2 of 458 ResNet-50 to ConvNeXt-V2-huge ($1\% \rightarrow 19.1\%$ UA2) demonstrates the effects of almost a decade 459 of CV advances, including self-supervised pretraining, hardware improvements, data augmentation, 460 architectural changes and new regularization techniques. More generally, we find a range of modern architectures and training strategies doing well (see Table 2, full results in Figure 8). This is gives a 461 positive view of how progress on standard CV benchmarks has tracked underlying robustness metrics, 462 contrasting with classical L_p adversarial robustness where standard training techniques have little 463 effect (Madry et al., 2017a). 464

 Scale, data augmentation and pretraining successfully improve unforeseen robustness. We do a more careful analysis of how three of the most effective CV techniques have improved robustness. As shown in Section 4.2, we find that data augmentation improves on unforeseen robustness, even in cases where they reduce standard accuracy. We compare the performance of ConvNeXt-V1 and ConvNeXt-V2 models, which differ through the introduction of self-supervised pretraining and a new normalization layer. When controlling for model capacity, the combination of this pre-training and normalisation layer demonstrates a large increase unforeseen robustness Table 8.

- 472
- 473 474

5 CONCLUSION

475

476 In this paper, we introduced a new benchmark for testing robustness against *unforeseen adversaries* 477 (ImageNet-UA) laying groundwork for continuing research in improving real world adversarial 478 robustness. We provide nineteen (eighteen novel) non- L_p attacks as part of our repository, using these 479 to construct a new metric UA2 (Unforeseen Adversarial Accuracy). We show that ImageNet-UA is 480 distinct from existing measures of robustness in the literature, and make use use it to evaluate classical 481 L_p training techniques—showing that the common practice of L_{∞} training and evaluation may be 482 misleading, as L_2 training shows higher unforeseen robustness. We additionally demonstrate that a 483 variety of interventions outside of L_p adversarial training can improve unforeseen robustness, both through existing techniques in the CV literature and through more specialised adversarial training 484 strategies. We hope that ImageNet-UA will be a useful tool as we continue to make progress towards 485 safer machine learning systems in real-world applications.

486 REFERENCES

493

499

505

- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.
- Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch.
 CoRR, abs/1712.09665, 2017. URL http://arxiv.org/abs/1712.09665.
- John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis* and machine intelligence, (6):679–698, 1986.
- 496 Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras,
 497 Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness.
 498 CoRR, abs/1902.06705, 2019. URL http://arxiv.org/abs/1902.06705.
- Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble
 of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216.
 PMLR, 2020.
- Francesco Croce and Matthias Hein. Adversarial robustness against multiple and single l_p -threat models via quick fine-tuning of robust classifiers, 2022.
- Sihui Dai, Saeed Mahloujifar, and Prateek Mittal. Formulating robustness against unforeseen attacks.
 arXiv preprint arXiv:2204.13779, 2022.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling CNNs with simple transformations. *arXiv preprint arXiv:1712.02779*, 2017.
- Alain Fournier, Don Fussell, and Loren Carpenter. Computer rendering of stochastic models.
 Commun. ACM, 25(6):371–384, June 1982. ISSN 0001-0782. doi: 10.1145/358523.358553. URL
 http://doi.acm.org/10.1145/358523.358553.
- Justin Gilmer, Ryan P. Adams, Ian J. Goodfellow, David Andersen, and George E. Dahl. Motivating the rules of the game for adversarial example research. *ArXiv*, abs/1807.06732, 2018.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial
 examples. arXiv preprint arXiv:1412.6572, 2014a.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial
 examples, 2014b. URL https://arxiv.org/abs/1412.6572.
- Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Jian Wang, Bing Yu, Wei Feng, and Yang Liu. Watch out! motion is blurring the vision of your deep neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 975–985. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/ file/0a73de68f10e15626eb98701ecf03adb-Paper.pdf.
- Qing Guo, Ziyi Cheng, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yang Liu, and Jianjun Zhao. Learning to
 adversarially blur visual object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10839–10848, 2021.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.
- 539 Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

540 541 542	Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. <i>arXiv preprint arXiv:1807.01697</i> , 2018.
543 544 545 546	Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In <i>Proceedings of the IEEE/CVF International Conference on Computer Vision</i> , pp. 8340–8349, 2021.
547 548 549	Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse Gabor convolution. <i>ACM Trans. Graph.</i> , 28(3):54:1–54:10, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531360. URL http://doi.acm.org/10.1145/1531326.1531360.
550 551 552	Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. <i>Advances in neural information processing systems</i> , 32, 2019.
553 554	Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models, 2020. URL https://arxiv.org/abs/2006.12655.
555 556 557	Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. <i>Advances in Neural Information Processing Systems</i> , 33:3487–3498, 2020.
558 559 560 561	Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In <i>Proceedings of the</i> <i>IEEE/CVF international conference on computer vision</i> , pp. 10012–10022, 2021.
562 563 564	Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pp. 11976–11986, 2022.
565 566	Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Learning to generate noise for multi-attack robustness, 2021a.
568 569	Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Learning to generate noise for multi-attack robustness. In <i>International Conference on Machine Learning</i> , pp. 7279–7289. PMLR, 2021b.
570 571 572	Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. <i>arXiv preprint arXiv:1706.06083</i> , 2017a.
573 574 575 576	Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017b. URL https://arxiv. org/abs/1706.06083.
576 577 578	Pratyush Maini, Eric Wong, and J. Zico Kolter. Adversarial robustness against the union of multiple perturbation models, 2020.
579 580 581	Karttikeya Mangalam, Haoqi Fan, Yanghao Li, Chao-Yuan Wu, Bo Xiong, Christoph Feichtenhofer, and Jitendra Malik. Reversible vision transformers. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pp. 10830–10840, 2022.
582 583 584 585	Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. <i>Advances in Neural Information Processing Systems</i> , 34:3571–3583, 2021.
586 587	Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks. <i>arXiv preprint arXiv:1511.04599</i> , 2015.
588 589 590 591	Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. <i>arXiv preprint arXiv:2304.07193</i> , 2023.
592 593	Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In <i>Proceedings of the 2017 ACM on Asia conference on computer and communications security</i> , pp. 506–519. ACM, 2017.

594 595	Ken Perlin. Making noise, 1999. URL http://www.noisemachine.com/talk1/index.html, 2005.
596	Vinay Uday Prabhu. The blood diamond effect in neural art : On ethically troublesome images of the
597	imagenet dataset. 2019.
598	Haonan Oiu Chaowei Xiao Lei Yang Xinchen Yan Honglak Lee and Bo Li Semanticady:
599	Generating adversarial examples via attribute-conditional image editing. ArXiv. abs/1906.07927.
600	2019.
601	
602	Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
603	Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual
604 605	models from natural language supervision. In <i>International conference on machine learning</i> , pp. 8748–8763. PMLR, 2021.
606	Laslie Dies Erie Wang and Zies Kalter Overfitting in adversarially reduct deer learning. In
607 608	International Conference on Machine Learning, pp. 8093–8104. PMLR, 2020.
600	Ali Shahin Shamsabadi. Changiae Oh, and Andrea Cavallaro. Semantically adversarial learnable
610	filters. <i>IEEE Transactions on Image Processing</i> , 30:8075–8087, 2021.
611	Richard Shin and Dawn Song. JPEG-resistant adversarial images. In NIPS 2017 Workshop on
612	Machine Learning and Computer Security, 2017.
613	
614	Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial
615	examples with generative models. In <i>NeurIPS</i> , 2018.
617	Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit,
619	and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision
610	transformers. Transactions on Machine Learning Research, 2022. ISSN 2835-8856. URL
620	https://openreview.net/forum?id=4nPswr1KcP.
621	David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generaliz-
622	ing to unseen attacks. In International Conference on Machine Learning, pp. 9155–9166. PMLR,
623	2020.
624	Chiefe Charles I. Which I. Zhanda, H. Chiefe Landa, Landa, David E. F. Landa, Cardfill
625	Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Ernan, Ian Goodfellow, and Pob Fargus. Intriguing properties of neural networks. 12 2013
626	and Rob Pergus. Intriguing properties of neural networks. 12 2015.
627	Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations.
628	Advances in neural information processing systems, 32, 2019.
629	Elerion Tramèr Descel Dunrá Gili Dusek, Gianzarle Dellegrine, and Dan Bonch. Ad versariely
630	Defeating percentual ad-blocking CoRR abs/1811 03194 2018 LIRL http://arxiv.org/
631	abs/1811.03194.
632	
633	Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations
634	by penalizing local predictive power. Advances in Neural Information Processing Systems, 32,
635	2019.
636	Zekai Wang Tianyu Pang Chao Du Min Lin Weiwei Liu and Shuicheng Yan Better diffusion
637	models further improve adversarial training. 2023.
638	
639	Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and
640	Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach.
641	arxiv preprint arXiv:1801.10578, 2018.
642	Sanghyun Woo, Shoubhik Debnath, Ronghang Hu. Xinlei Chen, Zhuang Liu, In So Kweon, and
643	Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. <i>arXiv</i>
644	preprint arXiv:2301.00808, 2023.
645	
646	Chaowei Alao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially trans- formed adversarial examples. In <i>Proceedings of (ICLP) International Conference on Learning</i>
647	Representations, April 2018.

Yaodong Yu, Zitong Yang, Edgar Dobriban, Jacob Steinhardt, and Yi Ma. Understanding generalization in adversarial training via the bias-variance decomposition. *arXiv preprint arXiv:2103.09947*, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan.
Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/ zhang19p.html.

Zhengyu Zhao, Zhuoran Liu, and Marisa Larson. Towards large yet imperceptible adversarial image
 perturbations with perceptual color distance. *ArXiv*, abs/1911.02466, 2019.

702 A HYPERPARAMETERS 703

704	A.1	TRAINED MODELS
706	To ru	in our evaluations, we train a range of our own models to benchmark with:
707		
708		• CIFAR-10 WRN-28-10 robust models and TRADES models are respectively trained with the
709		official code of Rice et al. (2020) and Zhang et al. (2019) with the default hyperparameters
710		settings
/11		• The PAI-VR models on ImageNet100 were trained using the official code from Dat et al. (2022) and employed the hyperparameter settings outlined in the code of L aidlaw et al.
712		(2022) and employed the hyperparameter settings outlined in the code of Launaw et al. (2020).
714		• ImageNet100 DINOv2 Oquab et al. (2023) models are trained by finetuning a linear classifi-
715		cation head on the ImageNet100 dataset. We used a SGD optimizer with learning rate of
716		0.001 and employed early-stopping.
717		
718	A.2	Model Reference
719	We u	se a range of baseline models provided by other works, with model weights available as part of
720	their	open source distribution:
721		
723		• ImageNet
724		- ConvNeXt models are from Liu et al. (2022)
725		- ConvNeXt-V2 models are from Woo et al. (2023)
726		- ViT models are from Steiner et al. (2022)
727		- Swin models are from Liu et al. (2021)
728		- Reversible-ViT models are from Mangalam et al. (2022)
729		- CLIP (ViT-L/14) is from Radford et al. (2021)
730		– DINOv2 models are from Oquab et al. (2023)
731		- MAE models are from He et al. (2022)
732		• CIFAR-10
733		- WideResNet TRADES models are from Zhang et al. (2019)
734		- WRN + Diffusion models are from Wang et al. (2023)
735		- Meta noise models are from Madaan et al. (2021b)
737		- ResNet50 VR models are from Dai et al. (2022)
738		- ReColorAdv models are from Laidlaw & Feizi (2019)
739		- StAdy modesl are from Xiao et al. (2018)
740		– Multi attack models are from Tramèr et al. (2018)
741		- The Multi steepest descent model is from Maini et al. (2020)
742		– PAT models are from Laidlaw et al. (2020)
743		- Pre-trained ResNet18 L_{∞} , L_2 and L_1 models are from Croce & Hein (2022)
744		• ImageNet100
745		BasNet50 DAT models are from Leidley, et al. (2020)
746		- ResNet50 PAT models are from Dai et al. (2020)
7/9		- Reside D PAT + VR models are from Dat et al. (2022)
749		- DINOV2 models are from Oquab et al. (2023)
750	٨ 2	ATTACK DADAMETEDS
751	А.Э	AT TACK TAKAMETEKS
752	To er	sure that our attacks are maximally effective, we perform extensive hyper-parameter sweeps to

		Step Size	Num Steps	Low Distortion	Medium Distortion	High Distortion	Distance Metric
	PGD	0.004	50	2/255	4/255	8/255	L_{∞}
	Gabor	0.0025	100	0.02	0.04	0.06	L_{∞}
	Snow	0.1	100	10	15	25	L_2
	Pixel	1	100	3	5	10	L_2
Core Attacks	JPEG	0.0024	80	1/255	3/255	6/255	L_{∞}
	Elastic	0.003	100	0.1	0.25	0.5	L_2
	Wood	0.005	80	0.03	0.05	0.1	L_{∞}
	Glitch	0.005	90	0.03	0.05	0.07	L_{∞}
	Kaleidoscope	0.005	90	0.05	0.1	0.15	L_{∞}
	Edge	0.02	60	0.03	0.1	0.3	L_{∞}
	FBM	0.006	30	0.03	0.06	0.3	L_{∞}
	Fog	0.05	80	0.3	0.5	0.7	L_{∞}
	HSV	0.012	50	0.01	0.03	0.05	L_{∞}
	Klotski	0.01	50	0.03	0.1	0.2	L_{∞}
extra Attacks	IVIIX Dalardat	1.0	70	J 1	10	40	L_2
	Pokadot	0.3	70	1	3	5	L_2
	PHSOI	0.0015	30 40	0.01	0.05	0.1	L_{∞}
	Diul Texture	0.05	40	0.1	0.5	0.0	L_{∞}
	TEXTUIC	0.00075	00	0.01	40	100	L_{∞}
	Whirlpool	4.0 Table	40 10: Attack	parameters f	or CIFAR-10-UA	100	L ₂
	Whirlpool	4.0 Table	40 10: Attack	parameters f	or CIFAR-10-UA		
	Whirlpool	4.0 Table Step Size	40 10: Attack Num Steps	to parameters f	or CIFAR-10-UA	High Distortion	Distance Metric
	Whirlpool PGD	4.0 Table Step Size	40 10: Attack Num Steps 50	parameters f Low Distortion 2/255	40 or CIFAR-10-UA Medium Distortion	High Distortion 8/255	L_2 Distance Metric L_{∞}
	Whirlpool PGD Gabor Snow	4.0 Table Step Size 0.008 0.0025 0.2	40 10: Attack Num Steps 50 80 20	10 parameters f Low Distortion 2/255 0.02	40 or CIFAR-10-UA Medium Distortion 4/255 0.03	High Distortion 8/255 0.04	L_2 Distance Metric L_{∞} L_{∞}
	Whirlpool PGD Gabor Snow Piyal	4.0 Table Step Size 0.008 0.0025 0.2	40 10: Attack Num Steps 50 80 20 60	Low Distortion 2/255 0.02 3	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5	High Distortion 8/255 0.04 5	L_2 Distance Metric L_{∞} L_2 L_2
Core Attacks	Whirlpool PGD Gabor Snow Pixel IPEG	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024	40 10: Attack Num Steps 50 80 20 60 50	Low Distortion 2/255 0.02 3 1 1/255	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255	High Distortion 8/255 0.04 5 10 6/255	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006	40 10: Attack Num Steps 50 80 20 60 50 30	10 2 parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25	High Distortion 8/255 0.04 5 10 6/255 0.5	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_2
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625	40 10: Attack Num Steps 50 80 20 60 50 30 70	IO c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1 0.03	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_2 L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.00625 0.000625 0.0025	40 10: Attack Num Steps 50 80 20 60 50 30 70 60	IO c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.03	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.1	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_2 L_{∞} L_2 L_{∞} L_2 L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625 0.0005	40 10: Attack Num Steps 50 80 20 60 50 30 70 60 30	IO Image: construction 2/255 0.02 3 1 1/255 0.01 0.03 0.05	40 or CIFAR-10-UA <u>Medium Distortion</u> 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.1 0.15	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_{∞} L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.00625 0.005 0.005 0.05	40 10: Attack Num Steps 50 80 20 60 50 30 60 60	IO Low Distortion 2/255 0.02 3 1 1/255 0.03 0.03 0.05	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.1 0.1 0.15 0.3	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_2 L_2 L_{∞} L_{∞} L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625 0.0025 0.0025 0.005 0.02 0.006	40 10: Attack Num Steps 50 80 20 60 50 30 60 30 60 30	IO c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1 0.03 0.05 0.03 0.02	40 or CIFAR-10-UA <u>4/255</u> 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_{∞} L_{∞} L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625 0.0025 0.0025 0.005 0.002 0.006 0.006 0.006 0.006 0.005	40 10: Attack Num Steps 50 80 20 60 50 30 70 60 30 60 30 40	10 a parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1 0.03 0.03 0.02 0.3	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04 0.4	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08 0.5	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_{∞} L_{∞} L_{∞} L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.0025 0.0025 0.0025 0.0025 0.002 0.006 0.005 0.003	40 10: Attack Num Steps 50 80 20 60 50 30 70 60 30 60 30 40 20	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.03 0.05 0.03 0.02 0.3 0.01	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04 0.4 0.02	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08 0.5 0.03	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_{∞} L_{∞} L_{∞} L_{∞} L_{∞} L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625 0.005 0.005 0.055 0.005	40 10: Attack Num Steps 50 80 20 60 30 70 60 30 60 30 60 30 60 30 60 30 50 50 50 50 50 50 50 50 50 5	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1 0.03 0.05 0.03 0.02 0.3 0.01 0.03	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04 0.4 0.02 0.05	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.1	$\begin{array}{c} L_2\\ \hline \text{Distance Metric}\\ L_{\infty}\\ L_2\\ L_2\\ L_2\\ L_{\infty}\\ L_2\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty} \end{array}$
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski Mix	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.000625 0.0025 0.0025 0.0025 0.005 0.005 0.005 0.003 0.005 0.5	40 10: Attack Num Steps 50 80 20 60 50 30 60 30 60 30 60 30 40 20 50 30 50 30 30 40 50 30 30 30 30 30 30 30 30 30 3	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.1 0.03 0.05 0.03 0.02 0.3 0.03 0.03 0.03 0.03 0.03 0.03 1	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04 0.4 0.02 0.05 5	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.03 0.03 0.1 10	$\begin{array}{c} L_2\\ \hline \text{Distance Metric}\\ L_{\infty}\\ L_{\infty}\\ L_2\\ L_2\\ L_2\\ L_{\infty}\\ L_{2}\\ \end{array}$
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski Mix Pokadot	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0025 0.0025 0.0025 0.0025 0.0025 0.0025 0.005 0.003 0.003 0.005 0.3 0.3	40 10: Attack Num Steps 50 80 20 60 50 30 60 30 60 30 40 20 60 30 40 20 60 30 40 20 60 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 20 60 50 30 40 50 50 50 50 50 50 50 50 50 5	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.03 0.03 0.05 0.03 0.02 0.3 0.01 0.03 1 1	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.1 0.1 0.04 0.4 0.02 0.05 5 2 2	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.1 10 3 0.1 10 3 0.1 10 3 0.1 10 10 10 10 10 10 10 10 10 1	$\begin{array}{c} L_2\\ \hline \\ \textbf{Distance Metric}\\ L_{\infty}\\ L_2\\ L_2\\ L_2\\ L_2\\ L_2\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{2}\\ L_{2}\\ L_{2}\\ \end{array}$
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski Mix Pokadot Prison	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.000625 0.0025 0.0025 0.005 0.005 0.002 0.006 0.005 0.003 0.005 0.3 0.0015 0.015 0.015	40 10: Attack Num Steps 50 80 20 60 50 30 60 30 60 30 40 20 50 30 40 20 20 20 20 20 20 20 20 20 2	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.03 0.03 0.02 0.3 0.01 0.01 0.01	40 or CIFAR-10-UA <u>4/255</u> 0.03 4 5 3/255 0.05 0.05 0.05 0.1 0.1 0.04 0.4 0.02 0.03 5 2 0.03 4 0.02 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.03 0.04 0.04 0.04 0.05 0.03 0.03 0.04 0.04 0.04 0.04 0.05 0.03 0.03 0.04 0.04 0.04 0.04 0.05 0.05 0.05 0.05 0.05 0.05 0.05 0.02 0.03 0.04 0.04 0.04 0.05 0	High Distortion 8/255 0.04 5 10 6/255 0.1 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.1 10 3 0.1	L_2 Distance Metric L_{∞} L_2 L_2 L_2 L_2 L_{∞}
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski Mix Pokadot Prison Blur	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.000625 0.0025 0.0025 0.0025 0.0025 0.0025 0.005 0.005 0.003 0.005 0.3 0.0015 0.0015 0.005	40 10: Attack Num Steps 50 80 20 60 50 30 70 60 30 60 30 40 20 50 30 40 20 50 30 20 60 30 20 50 30 20 60 30 20 50 30 20 50 30 20 50 30 20 50 30 20 50 30 20 50 30 20 50 30 20 50 30 20 20 20 20 20 20 20 20 20 2	10 c parameters f Low Distortion 2/255 0.02 3 1 1/255 0.03 0.03 0.02 0.3 0.01 0.03 0.01 0.01 0.01	40 Medium Distortion 4/255 0.03 4 5 3/255 0.05 0.1 0.04 0.4 0.02 0.05 5 2 0.03 0.3 0.3	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.1 10 3 0.1 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2	$\begin{array}{c} L_2\\ \hline \text{Distance Metric}\\ L_{\infty}\\ L_2\\ L_2\\ L_2\\ L_{\infty}\\ L_2\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{2}\\ L_{2}\\ L_{2}\\ L_{2}\\ L_{\infty}\\ L_{\infty$
Core Attacks	Whirlpool PGD Gabor Snow Pixel JPEG Elastic Wood Glitch Kaleidoscope Edge FBM Fog HSV Klotski Mix Pokadot Prison Blur Texture Whielengel	4.0 Table Step Size 0.008 0.0025 0.2 1.0 0.0024 0.006 0.0025 0.0025 0.0025 0.0025 0.0025 0.0025 0.005 0.005 0.003 0.005 0.3 0.0015 0.015 0.003 16.0 16	40 10: Attack Num Steps 50 80 20 60 50 30 70 60 30 60 30 60 30 60 30 40 20 50 30 20 60 30 20 50 50 50 50 50 50 50 50 50 5	10 c parameters f Low Distortion 2/255 0.02 3 1/255 0.1 0.03 0.05 0.03 0.01 0.03 0.01 0.01 0.1 0.01 0.01 0.01 0.01 0.01 0.01 0.01	40 or CIFAR-10-UA Medium Distortion 4/255 0.03 4 5 3/255 0.25 0.05 0.05 0.1 0.1 0.04 0.02 0.05 5 2 0.03 0.3 0.1 100	High Distortion 8/255 0.04 5 10 6/255 0.5 0.1 0.1 0.15 0.3 0.08 0.5 0.03 0.1 10 3 0.1 0.6 0.200	$\begin{array}{c} L_2\\ \hline \\ \text{Distance Metric}\\ L_{\infty}\\ L_2\\ L_2\\ L_2\\ L_{\infty}\\ L_2\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{\infty}\\ L_{2}\\ L_{2}\\ L_{2}\\ L_{\infty}\\ L_$

Table 9: Attack parameters for ImageNet-UA

792 793

794

796

797

798

799

800

756

B DESCRIPTIONS OF THE 11 ADDITIONAL ATTACKS.

Blur. Blur approximates real-world motion blur effects by passing a Gaussian filter over the original image and then does a pixel-wise linear interpolation between the blurred version and the original, with the optimisation variables controlling the level of interpolation. We also apply a Gaussian filter to the grid of optimisation variables, to enforce some continuity in the strength of the blur between adjacent pixels. This method is distinct from, but related to other blurring attacks in the literature (Guo et al., 2020; 2021).

Edge. This attack functions by applying a Canny Edge Detector (Canny, 1986) over the image to locate pixels at the edge of objects, and then applies a standard PGD attack to the identified edge pixels.

Fractional Brownian Motion (FBM). FBM overlays several layers of Perlin noise (Perlin, 2005) at different frequencies, creating a distinctive noise pattern. The underlying gradient vectors which generate each instance of the Perlin noise are then optimised by the attack.

Fog. Fog simulates worst-case weather conditions, creating fog-like occlusions by adversarially
 optimizing parameters in the diamond-square algorithm (Fournier et al., 1982) typically used to
 render stochastic fog effects.

839

840

841



Figure 5: **The full suite of attacks.**We present nineteen differentiable non-Lp attacks as part of our codebase . For the purpose of visualization, higher distortion levels than are used in our benchmark have been chosen. See Appendix G for adversarial examples generated with the distortion levels used within our benchmark, and Appendix K for a human study on semantic preservation

HSV. This attack transforms the image into the HSV color space, and then optimises PGD in that latent space. Due to improving optimisation properties, a gaussian filter is passed over the image.

Klotski. The Klotski attack works by splitting the image into blocks, and applying a differentiable
translation to each block, which is then optimised.

Mix. The Mix attack functions by performing differntiable pixel-wise interpolation between the original image and an image of a different class. The level of interpolation at each pixel is optimised, and a gaussian filter is passed over the pixel interpolation matrix to ensure that the interpolation is locally smooth.

Polkadot. Polkadot randomly selects points on the image to be the centers of a randomly coloured circle, and then optimising the size of these circles in a differentiable manner.

Prison. Prison places grey "prison bars" across the image, optimising only the images within the
prison bars. This attack is inspired by previous "patch" attacks (Brown et al., 2017), while ensuring
that only the prison bars are optimised.

857 Texture. Texture works by removing texture information within an images, passing a Canny Edge
858 Detector (Canny, 1986) over the image to find all the pixels which are at the edges of objects, and
859 then filling these pixels in black—creating a silhouette of the original image. The other non-edge (or
860 "texture") pixels are then whitened, losing the textural information of the image while preserving the
861 shape. Per-pixel optimisation variable control the level of whitening.

Whirlpool. Whirlpool translates individual pixels in the image by a differentiable function creating
 a whirlpool-like warpings of the image, optimising the strength of each individual whirlpool.

⁸⁶⁴ C Full Description of Wood Attack

In Figure 3, we give a high-level explanation of the Wood attack. Here, we give a more detailed explanation of this figure.

868 Given a classifier f, the Wood attack with distortion level ε functions by taking a set of adversarial latent variables $\delta_n \in \mathbb{R}^{m \times m \times 2}$ (representing a vector field of per-pixel displacements), applies 870 $project_{\varepsilon}^{\varepsilon}$ to project this field into the ε ball in the L_{p} metric and then uses bi-linear interpola-871 tion to upsample the latent variables to the input image size. The upsampled latent variables are 872 then used to make the wood noise, by using an element-wise mapping $F: \mathbb{R}^{n \times n \times 2} \to \mathbb{R}^{n \times n}$, 873 taking a coordinate to the (power of) the sine of its distance from the center of the image i.e. 874 $F(I) = \sin(\sqrt{(X)^2 + (Y)^2})^{\beta}$, where $X_{ij} = I_{ij0} - n/2$ and $Y_{ij} = I_{ij1} - n/2$ and β is an attack hyperparameter. When applied to constant coordinate tensor $C \in \mathbb{R}^{n \times n \times 2}$, $C_{ij} = (i, j)$, this func-875 876 tion creates the distinctive "wood rings" of the Wood attack, which are then multiplied with the 877 input image to produce adversarial input. By virtue of the differentiability of this process, we can 878 backpropagate through this noise generation and optimize the adversarial image x_{adv} by performing PGD (Madry et al., 2017a) on the input latent variables. 879

880

882

899

900 901

902

903 904

905 906 907

908

909

910

911

912 913 914

881

D PROCESS FOR DESIGNING ATTACKS AND SELECTING CORE ATTACKS

Our design of attacks is guided by two motivations: defending against unforeseen adversaries and robustness to long-tail scenarios. Unforeseen adversaries could implement novel attacks to, e.g., evade automated neural network content filters. To model unforeseen adversaries that might realistically appear in these scenarios, we include digital corruptions similar to what one might see on YouTube videos trying to evade content filters. These include attacks such as Kaleidoscope and Prison. To model long-tail scenarios, we include worst-case versions of common corruptions, like JPEG, Snow, and Fog.

In preliminary experiments, we found that some of these attacks were more effective than others, 890 leading to lower accuracy with fewer steps. We also found that performance on some attacks was 891 correlated between models. For example, both "Prison" and "Edge", are pixel-level attacks, so 892 robustness to one was correlated with robustness to the other. To increase the diversity and efficiency 893 of our evaluation, we selected a core set of eight attacks based on their effectiveness and diversity, 894 considering both visual diversity and the accuracy profiles of different models. This was an iterative 895 process that led us to make substantial changes to some attacks. For example, we modified the 896 implementation of the Elastic attack to use larger, lower-frequency distortions, which maintained its 897 effectiveness while reducing correlation with PGD. 898

E ATTACK COMPUTATION TIME

We investigate the execution times of our attacks, finding that most attacks are not significantly slower than an equivalent PGD adversary.



Figure 6: Evaluation time of the attacks on the ImageNet test set using a ResNet50 model with batch size of 200 on a single A100-80GB GPU, Attack hyper-parameters are as described in Appendix A.

916 917

918 F FULL RESULTS OF MODEL EVALUATIONS 919

We benchmark a large variety of models on our dataset, finding a rich space of interventions affecting unforeseen robustness.

Pixe

33.2

32.9

31.5

28.5

27.8

32.9

27.6 30.1 21.3 29.4 26.0

26.0 16.6 23.7 33. 19.2 26.0 26.0

- 26.0 - 16.5 - 17.2 - 18.9 - 20.4 - 15.1 - 16.4 - 20.9 - 20.9

- 11.9 - 11.6 - 11.7 - 11.5

43 0

50

1000

50

1000

29.4

33.2

58.4

72.0

71 5

47.8

52.5 38.6 48.6

48.5

54.6

46.4

33

36.6

51.4

26.2

36.3

28.

37.6 15.1 34.1 30.5 31.3 31.5 16.2 35.0

10.7 16.3 21.0 31.5 32.0

21.5

50

10.7

11.5

35.7

37.6

38.3

33.0

46

41 7

36.1

37.5

36.5

38.9

35.8

38.3

27.0

18.3 36.

47. 18.9 37.5

37. 37. 31.6 30.7

31.4

31.4 15.1 15.8 22.6 13.7 31.9 27.1 23.3 28.8

- 7.2

100 0

50

100

- 31 - 30 - 17.6 - 13.5

31.7

44 5

F.1 IMAGENET

920

921

922 923

924

VIT-base Patch16 + L₊ 4/255 VConVNEXt-base + L₊ 4/255 DINO2 VIT-large Patch14 Swih-base ImageNet1K + L₊ 4/255 VIT-small Patch16 + L₊ 4/255 Swih-amall ImageNet1K + L₊ 4/255 ConVNEXt-V2-large ImageNet1K+22K VIT-base Patch16 Swih-base ImageNet1K ResNet50 + L₂ 3 MAE VIT-base Patch16 Swih-base ImageNet1K VIT-base Patch16 ImageNet1K+22K VIT-base Patch16 Swih-base ImageNet1K VIT-base Patch16 Swih-base ImageNet1K VIT-base Patch16 Swih-Base ImageNet1K ConVNEXt-base ImageNet1K ConVNEXt-base ImageNet1K ConVNEXt-base ImageNet1K+22K Swih-Site1 - 2755 ResNet50 + 1, 2755 ResNet50 925 Clean acc PGD (2/255) UA2 JPEG Elasti Wood Glitch Kaleidos 76.8 68.3 46.8 70.8 55.5 28.1 11.6 3.6 926 76.1 68.1 43.8 53.1 55.1 25.1 67.6 86.1 16.7 43.4 17.3 61.3 52.2 24.5 68. 10.7 2.2 6.3 44.6 55.5 42.5 44.0 75.0 72.8 74.1 73.4 927 64.9 41.9 50.8 53.6 20.4 63.5 41.7 49.8 51.5 $\begin{array}{c} = 22.4\\ = 12.6\\ = 130.6\\ = 130.6\\ = 130.7\\ = 32.4, = 122.4\\ = 22.4, = 122.4\\ = 22.4, = 122.4\\ =$ 52.6 49.9 46.6 35.0 41.1 33.7 41.0 42.5 40.9 38.8 50.6 928 46.5 55.0 58.0 50.4 37.8 36.2 34.0 33.0 929 42 44 35.4 26.8 25.4 0.9 2.2 3.4 0.3 85.1 54.2 930 33.0 35.4 32.8 50.8 931 86.6 31.3 47.3 41.8 84.3 12.0 30.6 46.0 36.3 54 0.4 0.4 15.6 18.3 0.5 50 0.5 14.6 0.6 0.6 0.6 0.6 10.6 0.6 10.6 22.8 0.8 24.7 14.4 37.7 11.4 0.8 37.7 0.7 0.7 932 62.8 29.4 33.5 30.5 48.3 - 3.8 - 3.2 1.1 29.2 39. 933 28.7 52.0 38.0 29.2 28.6 -0.4 -7.8 -1.3 -1.4 45.6 37.8 47.8 51.9 28.5 26.9 26.9 26.6 26.4 - 199 - 199 - 112 - 8 8 934 84.9 84.3 935 33.6 70.4 42.0 26.3 32.9 936 ConvNeXt-base ImageNet1K+22K ResNet50 + L_{ω} 4/255 ResNet50 + L_{ω} 8/255 85.8 0.1 26.3 0.3 43.4 о... 36.. 34.9 0.3 63.9 26.1 33.1 52.2 937 54.5 46.8 25.2 24.6 30.4 ResNet50 + L_ 8/255 ConvNexV-22-1my ImageNet1K+22K ResNet50 + L_ 1/255 Swin-smail ImageNet1K ConvNexX-base ImageNet1K MAE VIT-base Patch16 ConvNexX-v2-tiny ImageNet1K+22K ConvNexX-v2-tiny ImageNet1K+22K ResNet50 + L_ 0.5 ResNet50 + L_ 0.5/255 ConvNexX-smail ImageNet1K 38.5 33.0 43.4 83.9 72.0 0.1 24.6 24.5 24.2 23.8 23.3 42.9 938 2.2 83.2 0.9 0.3 0.2 0.6 47.3 41.9 41.5 939 83. 84. 940 39.6 73.2 84.5 30.1 1.4 1.7 941).z 31.1 44. 91. 92. 95. 16.8 92.5 12.6 93. 12.6 93. 12.6 93. 12.6 93. 12.6 93. 12.6 93. 12.6 93. 12.6 93. 12.5 94.4 92.5 94.4 94.4 92.5 94.4 82.7 0.5 44 5 73.7 31.0 ResNet50 + L_ 0.5/255 comvRex5-amil ImageNet1X ComvNeXV-2-nano ImageNet1X Reversible VT-base ComvNeXX-V2-nano ImageNet1X+2 ComvNeXX-Imy ImageNet1X VT-base Patch16 ImageNet1X Reversible VT-small ComvNeXX-ty-pico ImageNet1X+22X CuPV (VT-JL4) VT-small Patch16 ImageNet1X+22X 32.2 942 -0.6 -1.3 -8.2 -0.1 -0.2 -4.4.9 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.4 -0.1 -2.8 -1.5 -1.5 -1.5 -1.5 -1.5 -1.5 -1.2 -0.2 -0.4 -0.2 -0.1 -0.2 -0.4 -0.1 -0.2 -0.4 -0.2 -0.1 -0.2 -0.1 -0.2 -0.1 -0.1 -0.2 -0.1 -0.2 -0.1 -0.2 -0.1 -0.2 45.0 38.1 943 37.9 33.5 42.4 37.7 26.8 et1K+228 944 945 33.1 946 82. 33.3 75.5 37.4 37.4 23.6 21.1 23.8 26.8 37.8 20.8 36.3 CLD (VII-244) VIT-small Patch16 imageNet1K VIT-small Patch16 imageNet1K ComVeXIV-2-temto imageNet1K ComVeXIV-2-temto imageNet1K ResNet50 + NVMix ResNet50 + NVMix ResNet50 + LopAugAugMix ResNet50 + LopAugAugMix ResNet50 + LopAugAugMix ResNet50 + LopAugMix ResNet50 + DeepAugAugMix ResNet50 + DeepAugAugMix ResNet50 + DeepAugAugMix ResNet50 + Nump ResNet50 + Nump ResNet50 + ResNet1K+22K ResNet50 + ResNet1K+22K ResNet50 + RandAug ResNet50 + AMT ResNet50 + AMT VIT-small Patch16 ImageNet1K 947 78.8 81.4 948 949 - 35.9 - 24.9 - 16.9 - 27.9 - 20.5 - 11.3 - 10.9 - 22.1 - 22.5 - 16.2 950 74.8 951 80.7 77.5 952 77.5 76.0 953 75.5 954 19.6 ł 955 50 50 50 50 50 50 1000 1000 1000 1000 1000 956

Figure 7: ImageNet UA2 performance under low distortion.

973 974 975 976 977 978 979 980 981 982 Kaleidoscope Clean acc PGD (4/255) UA2 JPEG Elastic Wood Glitch Pixel Gabo Snov DINOv2 VIT-large Patch14 VIT-base Patch16 + L_ 4/255 ConvNeXt-base + L_ 4/255 VIT-small Patch16 + L_ 4/255 Swim-base imageNetLX + L_ 4/255 ConvNeXt-V2-large imageNetLX + L_ 4/255 ConvNeXt-V2-huge imageNetLX + L_ 4/255 ConvNeXt-V2-huge imageNetLX DINOv2 VIT-base Patch14 DINOv2 VIT-base Patch14 983 86.1 15.3 27.7 14.3 42.6 39.1 17. 17.2 14.2 29.9 46.2 76.8 57.1 25.8 52.6 26.3 47.2 - 11.8 - 12.4 - 10.4 - 10.4 - 10.4 - 10.4 - 10.4 - 10.5 - 10.1 - 10.5 $\begin{array}{c} = 0.1 \\ = 0.1 \\ = 0.2 \\ = 0.2 \\ = 0.2 \\ = 0.2 \\ = 0.1 \\$ 11.9 27.1 30.2
 19.5

 12.8

 19.7

 12.8

 19.2

 10.3

 2.2.1

 2.3.1

 2.2.2

 2.3.3

 2.2.2

 2.3.1

 2.2.3

 2.3.4

 2.2.5

 2.3.1

 2.3.2

 2.4.4

 2.2.5

 2.2.5

 2.2.5

 2.2.6

 2.2.7

 2.2.7

 2.2.8

 2.2.9

 2.4.4

 2.2.9

 2.4.1

 2.2.1

 2.2.2

 2.3.3

 2.2.2

 2.3.4

 2.2.2

 2.3.4

 2.1.1

 2.2.2

 2.3.4

 2.2.2

 2.3.2

 2.3.2

 2.3.2

 2.3.2

 2.3.3

 3.3.3

 3.3.3

 3.3.2

 3.3.3
 984 76.1 22.3 39.0 23.8 47.9 58.0 $\begin{array}{c} = 9.7\\ = 7.9\\ = 7.5\\ = 7.5\\ = 7.5\\ = 8.3\\ = 15.5\\ = 6.5\\ = 6.5\\ = 6.5\\ = 6.5\\ = 6.5\\ = 6.5\\ = 6.9\\ = 0.10, 0, 0\\ = 0.10, 0\\ =$ 23.8 21.2 20.5 23.4 39.1 18.9 42.5 35.2 26.6 33.9 21.5 20.7 20.2 19.2 18.4 17.7
 42.1

 44.1

 45.6

 35.6

 22.7

 22.8

 22.4

 22.7

 22.8

 22.4

 22.7

 22.8

 22.8

 22.4

 22.5

 22.6

 22.7

 22.8

 22.8

 22.8

 22.8

 22.8

 22.8

 22.8

 22.8

 22.8

 22.9

 22.9

 22.4

 22.9

 22.1

 22.8

 22.9

 22.1

 22.1

 21.8

 12.9

 12.9

 12.9

 12.9

 12.9

 12.0

 22.1

 12.1

 12.2

 12.8

 13.5

 17.5

 13.5

 17.5
 51.6 54.4 51.8 - 26.7 - 21.9 - 21.9 - 4.0 - 0.7 - 4.6 - 10.4 - 2.2 - 4.3 - 11.2 - 1.5 - 1.5 - 1.7 - 1.0 - 1.1 - 1.5 - 1.7 - 1.0 - 1.1 - 1.5 - 1.7 - 1.0 - 1.3 - 1.5 - 1.7 - 1.6 - 0.8 - 1.0 - 1.5 - 1.5 - 1.7 - 1.6 - 1.6 - 0.8 - 1.0 - 1.5 - 1.5 - 1.7 - 1.7 - 1.6 - 0.8 - 1.0 - 1.5 - 1.5 - 1.7 - 1.5 - 1.5 - 1.7 - 1.6 - 0.8 - 1.0 - 1.5 - 1. 72.8 985 74.1 75.0 0.0 87.3 73.4 86.3 986 50.7 36.7 0.0 0.0 86.2 $\begin{array}{c} -1, -7, -2\\ -1, -1, -2\\ -1, -1, -2\\ -1, -1, -2\\ -1, -1, -2\\ -1,$ 987 10 -0.0 - 10.7 -0.0 -0.0 -0.0 84.3 11.4 DINOV2 VII-base Patch14 ConvNeXt-V2-base ImageNet1K+22K ConvNeXt-xlarge ImageNet1K+22K Swin-base ImageNet1K 33.9 988 0.0 87.0 33.5 85.3 0.0 31.4 989 ConvNeXt-V2-large ImageNet1K ResNet50 + L₂ 5 85.8 0.0 ConvNexV-V2-large imageNetIK ResNet30-1-4, and the set of the set 56.1 -0.0 -0.0 -0.0 990 0.0 0.0 0.0 85.8 62.8 85.8 85.8 991 992 0.1 84.3 0.0 84.9 993 83.2 0.0 54.5 38.9 994 83.9 0.0 83.8 82.9 84.6 1.0 995 996 0.0 82 997 81.8 63.9 998 CLIP (VIT-L/14) 75.5 0.3 ViT-base Patch16 ImageNet1K+22K 84.5 0.0 0.0 0.0 0.0 -0.0 -0.0 -0.0 -0.0 ConvNeXt-V2-nano ImageNet1K+22k 82.0 999 ConvNeXt-V2-nano ImageNetIK+. ConvNeXt-tiny ImageNetIK Swin-tiny ImageNetIK Reversible-VIT-base ConvNeXt-tiny ImageNetIK+22K ResNet50 + L_1 1 82.1 81.4 1000 82.9 70.4 80.3 1001 Nested to t_2 1 Convext.V2-pico ImageNet1K ResNet50 + L_{*} 2/255 Reversible-VIT-small ResNet50 + OutMix ResNet50 + Moex ResNet50 + PixMix Convext50 + PixMix 69 1 20.0 1002 79.8 78.6 0.0 0.5 0.5 1003 79.0 0.0 0.0 0.0 78.1 ConvNeXt-V2-atto ImageNet1K 1004 76.7 ConvNeXIV2-atto ImageNetIK ViTbase Patchich GimageNetIK ResNet50 + L, 1/255 ResNet50 + Mirup ConvNeXIV2-femto ImageNetIK ViT-small Patch16 ImageNetIK ResNet50 + L2, 0.5 ResNet50 + L0, 0.5/255 ViT-small Patch16 ImageNetIK+22K ResNet50 + L0, 0.5/255 ViT-small Patch20 ImageNetIK+22K ResNet50 + AugMix ViT-base Patch20 ImageNetIK+22K 79.2 -0.0 -0.0 -0.0 -0.0 72.0 1005 1006 6. 1007 6.5 0.0 0.0 73.1 1008 76.7 77.5 0.0 ResNet50 + AugMix VT-base Patch32 ImageNet1K+22K VT-base Patch32 ImageNet1K ResNet50 + Stylked mageNet ResNet50 + L₂ 0.1 ResNet50 + L₂ 0.1 ResNet50 + ANT ResNet50 + ANT ResNet50 + VT-small Patch32 ImageNet1K+22K VT-tiny Patch16 ImageNet1K+22K 1009 0.0 0.0 0.1 0.0 0.0 0.0 1010 1011 1012 75.5 1013 100 50 1000 50 1000 50 1000 50 1000 50 1000 50 1000 50 1000 50 1000 50 50 100 1014 Figure 8: ImageNet UA2 performance under medium distortion 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025

1027										
1028										
1020										
1020										
1030										
1031										
1032										
1033										
1034										
1035										
1030		Clean acc PGD (8/255	5) UA2 JPE	G Elastic	Wood	Glitch	Kaleidoscop	e Pixel	Snow	Gabor
1037	DINOv2 ViT-large Patch14 ConvNeXt-V2-large ImageNet1K+22K	- 86.1 - 14.4 - 87.3 -0.0	- 19.2 - 13.1 - 10.5 - 0.0	- 23.5	- 25.4	- 14.7 - 16.7	- 29.4	- 12.1 - 4.7	-0.6	- 22.7
1038	DINOv2 ViT-base Patch14 ViT-base Patch16 + L _o 4/255 Swin-large ImageNet1K	- 84.3 - 11.0 - 76.8 - 32.4 - 86.3 -0.0	- 10.1 - 10.1 - 9.9 - 19.6 - 9.3 - 0.0	- 10.6	- 13.1 - 28.9 - 13.8	- 7.7	- 12.4 - 15.5 - 15.2	-1.1 -1.2	-0.7	- 8.8 - 9.9 - 24.9
1039	ConvNeXt-V2-huge ImageNet1K Swin-base ImageNet1K	- 86.3 -0.0 - 85.3 -0.0	- 9.0 - 0.0 - 8.5 - 0.0	- 25.9	- 8.9	- 17.9	-4.6	-0.6	-0.1 -0.4	- 13.7
1040	ConvNeXt-V2-base ImageNet1K+22K ConvNeXt-xlarge ImageNet1K+22K	- 86.8 -0.0 - 87.0 -0.0	- 8.4 - 0.0 - 8.0 - 0.0	- 17.6	-14.7	- 12.8	-13.9 -0.6	-12.9	-0.4 -0.1	- 15.3
1041	ConvNeXt-V2-large ImageNet1K MAE ViT-large Patch16 ConvNeXt-base + / 4/255	- 85.8 -0.0 - 86.0 -0.0 76.1 - 34.5	- 7.6 - 0.0 - 7.3 - 0.0 - 7.1 - 7.0	22.6	- 10.8	- 10.2 - 6.0 - 7.3	-0.7	-0.7	-0.2	- 13.5
1042	ViT-small Patch16 + L _a 4/255 ConvNeXt-large ImageNet1K+22K	72.8 - 26.8	- 7.1 - 13.5 - 7.1 - 0.0	-13.9	- 24.5	- 6.0	-14.8 -0.8	-0.2	-0.1	-3.4
1043	Swin-base ImageNet1K + L_ 4/255 ConvNeXt-small + L_ 4/255	75.0 - 25.0 74.1 - 30.1	- 7.0 - 6.7 - 6.4 - 12.0	-15.4 -13.4	- 28.6 - 27.2	- 6.3 - 4.7	-14.8 -0.9	-0.3 -0.2	-0.2 -0.1	-13.9 -12.8
1044	ViT-base Patch8 ImageNet1K+22K Swin-small ImageNet1K + L _a 4/255 ConvNetK base ImageNet1K + 23K	- 85.8 -0.0 - 73.4 - 25.2	-15.7 -0.0 -15.6 -16.6	-11.6 -13.7	- 9.4 - 24.6	-2.0	-14.6 -12.6 -0.2	-1.1 -0.2	-0.2	-16.7
1045	ConvNeXt-V2-base ImageNet1K ViT-large Patch16 ImageNet1K+22K		- 5.2 - 0.0 - 5.2 - 0.0	- 15.3	- 6.7	- 7.5	-2.3	-0.2	-0.1 -0.1	- 9.9
1046	ConvNeXt-large ImageNet1K Swin-small ImageNet1K	- 84.3 -0.0 - 83.2 -0.0	- 1 5.1 -0.0 - 1 4.9 -0.0	- 15.7 - 11.7	-9.1 -11.0	-3.8 -5.2	-1.3 -1.6	-0.3 -0.7	-0.1 -0.1	- 10.5 - 9.1
1047	ResNet50 + L ₂ 5 ConvNeXt-V2-tiny ImageNet1K+22K ConvNeXt-T2-K	56.1 - 13.8 83.9 -0.0	-14.9 -20.4 -14.6 -0.0	-11.6 -9.9	- 8.2 - 10.9 - 9 7	- 7.5	-0.2	-0.8	-0.0 -0.2	-0.5 -0.5 -0.1
1048	ConvNeXt-base ImageNet1K CLIP (VIT-L/14)		-4.4 -0.0 -4.3 -0.1	- 13.8	- 8.2 - 6.7	-4.1 -4.9	-0.7	-0.4	-0.1 -0.3	- 7.7
1049	ConvNeXt-V2-tiny ImageNet1K MAE ViT-base Patch16	- 82.9 -0.0 83.8 -0.0	-14.1 -0.0 -13.7 -0.0	- 10.5 - 12.2	- 8.3 - 5.3	-4.5 -2.0	-1.0 -1.0	-0.5 -1.2	-0.1 -0.0	- 8.1 - 8.2
1050	ConvNeXt-small ImageNet1K ResNet50 + L ₂ 3 ResNet50 + L 8055	- 83.1 -0.0 - 62.8 - 8.4	-B.6 -0.0 -B.5 -B.3.8	-1.2	- 8.2	-2.8	-0.3	-0.4	-0.0 -0.0	-0.5
1051	ConvNeXt-V2-nano ImageNet1K ConvNeXt-V2-nano ImageNet1K+22K	81.8 -0.0 82.0 -0.0	-3.2 -0.0 -3.2 -0.0	- 8.2	- 8.8	-13.5	-0.9	-0.8	-0.2 -0.1	-3.0 -3.6
1052	Reversible-ViT-base multiscale ViT-base Patch16 ImageNet1K+22K	82.7 -0.0 84.5 -0.0	-3.1 -0.0 -3.0 -0.0	- 7.7 - 15.2	- 7.0 - 13.9	-1.9 -1.5	-2.0 -13.4	-1.3 -0.4	-0.1 -0.0	- 4.9 - 9.3
1053	ConvNeXt-tiny ImageNet1K+22K ConvNeXt-tiny ImageNet1K	- 82.9 -0.0 - 82.1 -0.0	-13.0 -0.0 -12.9 -0.0	- 7.0	- 7.6	-12.6	-0.1	-0.5 -0.1	-0.2	-15.7 -14.9
1054	ResNet50 + PixMix ResNet50 + Moex		-2.9 -0.0 -2.8 -0.0 -2.7 -0.0	-13.6	- 7.2	-0.4	-0.0	-0.2	-0.0	- 11.2
1055	Reversible-ViT-base ResNet50 + CutMix	81.7 -0.0 78.6 -0.1	-12.7 -0.0 -12.7 -0.0	-15.6 -15.5	-14.8 9.4	-1.4 -1.6	-1.1 -0.1	-1.4 -13.0	-0.1 -0.6	- 7.2 -1.1
1056	Reversible-ViT-small ConvNeXt-V2-pico ImageNet1K	- 79.8 -0.0 - 80.3 -0.0	-12.5 -0.0 -12.5 -0.0	- 6.3 - 6.6	- 6.8	-0.9 -13.8	-0.4	-0.6 -0.2	-0.0 -0.0	-14.0
1057	ResNet50 + L _o 4/255 ResNet50 + DeepAug+AugMix	- 63.9 - 16.3 - 75.8 -0.0	-12.4 -0.1 -12.0 -0.0	-0.4	-10.8	- 5.7	-0.2	-0.2	-0.0 -0.0	-0.2
1058	ResNet50 + Mixup ResNet50 + Deepaugment	- 77.5 -0.0 - 76.7 -0.0	-12.0 -0.0 -12.0 -0.0	-13.2 -0.8	- 0 6.2 -1.5	-0.8 -0.3	-0.0 -0.0	-1.0 -0.1	-0.2 -0.0	-4.3 -13.0
1059	ResNet50 + L _o 2/255 ViT-base Patch16 ImageNet1K ConvNeXt V2 formto ImageNet1K	69.1 - 7.1 79.2 -0.0	-1.8 -0.0 -1.6 -0.0	-1.3 -13.5	- 9.2 - 3.5	-0.4	-0.1 -11.7	-0.1 -0.2	-0.0 -0.0	-0.3 -13.3 -1.5
1060	ResNet50 + L _a 1/255 VIT-small Patch16 ImageNet1K	72.0 -1.7 78.8 -0.0	-1.2 -0.1	-1.2	- 6.2 - 3.4	-1.9 -0.4	-0.1	-0.1	-0.0	-0.2
1061	ResNet50 + L ₂ 1 VIT-small Patch16 ImageNet1K+22K	70.4 -0.9 31.4 -0.0	-1.2 -1.4 -1.1 -0.0	-0.8 -2.0	-14.3 -1.7	-12.4 -0.5	-0.0 -0.6	-0.1 -0.1	-0.0 -0.0	-0.3 -14.1
1062	ViT-base Patch32 ImageNet1K+22K ResNet50 + L _x 0.5/255	- 80.7 -0.0 - 73.7 -0.1	-1.0 -0.0 -0.8 -0.0	-0.9	-1.4 -14.2	-0.8	-0.3	-0.1	-0.0	-0.3
1063	ResNet50 + AugMix ResNet50 + L ₂ 0.5		-0.8 -0.0 -0.7 -0.1	-0.9	-3.3 -3.1	-0.6 -1.4	-0.0	-0.1	-0.0	-1.0
1064	ResNet50 + Stylised ImageNet ResNet50 + ANT	76.7 -0.0 76.1 -0.0	-0.6 -0.0 -0.5 -0.0	-1.1 -0.4	-12.1 -0.7	-0.4 -0.3	-0.0 -0.0	-0.1 -0.1	-0.0 -0.0	-0.7 -2.3
1065	ResNet50 + L ₂ 0.1 ResNet50 + RandAug	74.8 -0.0	-0.4 -0.0 -0.4 -0.0	-0.7	-1.8 -1.5	-0.6 -0.2	-0.0	-0.1 -0.1	-0.0	-0.4 -0.8
1066	VIT-Small Patch32 ImageNet1K+22K ResNet50 VIT-tiny Patch16 ImageNet1K+22K	- 76.1 -0.0 - 75.5 -0.0	-0.3 -0.0 -0.2 -0.0	-0.7	-1.2	-0.2 -0.2	-0.0	-0.1	-0.0	-0.2
1067		0 100 0 50 :	1000 50 1000 50	1000 50	1000 50 1	.000 50 :	1000 50 1	1000 50	1000 50	1000 50 100
1068		Figure 0. I	mageNet UA	2 perform	nance ur	der hie	h distor	tion		
1069		Figure 9. 1	magenet UA	2 periorn	lance ui	luer mg	ii uistoi	tion		
1070										
1071										
1072										
1073										
1074										
1075										
1076										
1077										
1078										
1079										











1242 F.4 EXPLORING THE ROBUSTNESS OF DINOV2

Given the strong adversarial robustness of DINOv2 models under the PGD attack (Appendix F), we further evaluate the DINOv2 model under AutoAttack Croce & Hein (2020). Table 11 and Table 12 show that although for the robust ResNet50 model AutoAttack performs similarly to PGD, it is able to reduce the accuracy of DINOv2 models to 0.0% across all the distortion levels. Future work may benefite from applying the AutoAttack benchmark as a comparison point, instead of the base PGD adversary.

Table 11: Attacked accuracies of models on ImageNet

	ResNet50 + L_{∞} 8/255	DINOv2 ViT-base Patch14	DINOv2 ViT-large Patch14
PGD (2/255)	46.8%	12.0%	16.7%
APGD-CE (2/255)	46.2%	1.0%	1.0%
APGD-CE + APGD-T (2/255)	43.6%	0.0%	0.0%
PGD (4/255)	38.9%	11.4%	15.3%
APGD-CE (4/255)	37.9%	0.9%	0.8%
APGD-CE + APGD-T (4/255)	33.8%	0.0%	0.0%
PGD (8/255)	23.9%	11.0%	14.4%
APGD-CE (8/255)	22.6%	0.6%	0.7%
APGD-CE + APGD-T (8/255)	18.4%	0.0%	0.0%

Table 12: Attacked accuracies of models on ImageNet100

	ResNet50 + L_{∞} 8/255	DINOv2 ViT-base Patch14	DINOv2 ViT-large Patch14
PGD (2/255)	64.5%	34.3%	42.3%
APGD-CE (2/255)	64.4%	17.6%	20.0%
APGD-CE + APGD-T (2/255)	64.1%	0.0%	0.0%
PGD (4/255)	45.7%	32.6%	39.7%
APGD-CE (4/255)	45.2%	16.4%	17.3%
APGD-CE + APGD-T (4/255)	44.6%	0.0%	0.0%
PGD (8/255)	15.7%	31.5%	37.7%
APGD-CE (8/255)	14.7%	15.5%	14.5%
APGD-CE + APGD-T (8/255)	13.6%	0.0%	0.0%

1275

1250

1251

1262 1263

1264

1276

1277 F.5 PERFORMANCE VARIANCE

1278
1279As described in Section 3.2, we perform adversarial attacks by optimizing latent variables which are
randomly initialized in our current implementation, so the adversarial attack's performance can be
affected by the random seed for the initialization. To study the effect of random initializations, we
compute the UA2 performances of three samples of two ImageNet models, ResNet50 and ResNet50
+ L_2 5. We observe the standard deviations of UA2 of these two models across 5 different seeds to
be respectively 0.1% and 0.04% concluding that the variation in performance across the ImageNet
dataset is minor.

1287 1288

G IMAGES OF ALL ATTACKS ACROSS DISTORTION LEVELS

We provide images of all 19 attacks within the benchmark, across the three distortion levels.

- 1291
- 1292
- 1293
- 1294
- 1295

¹²⁸⁵ 1286







distortion (last row) on a standard ResNet50 model



distortion (last row) on a standard ResNet50 model





1566 H SCALING BEHAVIOUR OF OUR ATTACKS

To see how our attacks perform across model scale, we make use of the ConvNeXt-V2 model suite (Woo et al., 2023) to test the performance of our attacks as we scale model size. We find that capacity improves performance across the board, but find diminishing returns to simply scaling up the architectures, pointing towards techniques described in Section 4.2.



Figure 21: **Unforeseen Robustness across model scale.** We measure UA2 across model scale by evaluating the performance of ConvNeXt-V2 (Woo et al., 2023) models on ImageNet-UA, finding that scale improves performance, although the benchmark still provides a challenge to the largest models.

¹⁵⁹⁹ I DISTRIBUTION SHIFT COMPARED TO UNFORESEEN ROBUSTNESS

Table 13: Distribution-shift benchmarks and UA2 Comparing performance on ImageNet-Sketch and ImageNet-R to performance against both non-optimized and optimized versions of UA2. We observe that performance on standard distribution shift benchmarks is correlated with performance on non-optimized UA2, while optimized UA2 settings favor models which have been trained for worst-case settings.

Model	UA2 (non-optimised)	ImageNet-Sketch Acc.	ImageNet-R Acc.	UA2
Resnet 50	55.2	24.1	36.2	1.6
Resnet50 + AugMix	59.1	28.5	41.0	3.5
Resnet50 + DeepAug	60.2	29.5	42.2	3.0
Resnet50 + Mixup	59.9	26.9	39.6	4.8
Resnet50 + L_2 , ($\varepsilon = 5$)	43.2	24.2	38.9	13.9
Resnet50 + L_{∞} , ($\varepsilon = 8/255$)	40.6	18.6	34.8	10



Figure 22: **Behaviour of core attacks across model scale.** We see the performance of the eight core attacks across the ConvNeXt-V2 model suite, with performance on attacks improving with model scale.

BENCHMARKING NON- L_p Adversarial Training Strategies J

We wish to compare training strategies which have been specifically developed for robustness against both a variety of and unforeseen adversaries. To this end, we use Meta Noise Generation (Madaan et al., 2021b) as a strong multi-attack robustness baseline, finding that on CIFAR-10-UA this leads to large increases in robustness (Table 14). We also evaluate Perceptual Adversarial Training (Laidlaw et al., 2020) and Variational Regularization (Dai et al., 2022), two techniques specifically designed to achieve unforeseen robustness. We also evaluate combining PixMix and L_p adversarial training. All of these baselines beat L_p training.

Table 14: Comparing alternative training strategies to L_p baselines We demonstrate that models trained using Meta Noise Generation (MNG) (Madaan et al., 2021b) improve over L_p training baselines on CIFAR-10-UA.

Training	Clean Acc.	UA2
Standard	95.8	7.4
$L_{\infty}, \varepsilon = 8/255$	86.5	39.8
$L_2, \varepsilon = 2$	95.5	21.4
MNG	88.9	51.1

Meta Noise Generation (MNG) out-performs L_p baselines. We find that MNG, a technique original developed for multi-attack robustness shows a 11.3% increase in UA2 on CIFAR-10-UA, and PAT shows a 3.5% increase in UA2.

Table 15: Specialised Unforseen robustness training strategies. We see that ImageNet-UA PAT (Laidlaw et al., 2020) and PAT-VR (Dai et al., 2022)trained ResNet50s improve over L_p baselines. Selected L_p models are the best Resnet50s from the bench-marking done in Figure 8, and for computational budget reasons they are trained on a 100-image subset of ImageNet, constructured by taking every 10th class.

Training
Standard $L_{\infty}, \varepsilon = 8/255$ $L_2, \varepsilon = 4800/255$
PAT PAT-VR

Table 16: PixMix and L_p training. We compare UA2 performance on CIFAR-10 of models trained with PixMix and adversarial training. Combining PixMix with adversarial training results in large improvements in UA2, demonstrating an exciting future direction for improving unforeseen robustness. All numbers denote percentages, and L_{∞} training was performed with the TRADES algorithm.

1720	Model	Clean Acc.	UA2
1721	WRN-40-2 + PixMix	95.1	15.00
1723	WRN-28-10 + L_{∞} 4/255	89.3	37.3
1724	WRN-28-10 + L_{∞} 4/255 + PixMix	91.4	45.1
1725	WRN-28-10 + L_{∞} 8/255	84.3	41.4
1726	WRN-28-10 + L_{∞} 8/255 + PixMix	87.1	47.4
1/2/			

HUMAN STUDY OF SEMANTIC PRESERVATION Κ

Table 17: Results of user study. We run a user study on the 200 class subset of ImageNet presented as part of ImageNet-R (Hendrycks et al., 2021), assessing the multiple-choice classification accuracy of human raters, allowing raters to choose certain images as corrupted. We use 4 raters per label and take a majority vote, finding high classification accuracy across all attacks.

1735			
1736	Attack Name	Correct	Corrupted or Ambiguous
1737	Clean	95.4	4.2
1738	Elastic	92.0	2.0
1739	Gabor	93.4	4.0
1740	Glitch	80.2	16.0
1741	JPEG	93.4	0.6
17/19	Kaleidescope	93.0	6.2
1742	Pixel	92.6	1.8
1743	Snow	90.0	3.2
1744	Wood	91.4	1.8
1745	Adversarial images average	01.2	4.5
1746	Auversallar lillages average	91.2	7.5

We ran user studies to compare the difficulties of labeling the adversarial examples compared to the clean examples. We observe that under our distribution of adversaries users experience a 4.2% drop in the ability to classify. This highlights how overall humans are still able to classify over 90% of the images, implying that the attacks have not lost the semantic information, and hence that models still have room to grow before they match human-level performance on our benchmark.

In line with ethical review considerations, we include the following information about our human study:

- How were participants recruited? We made use of the surgehq.ai platform to recruit all participants.
- How were the participants compensated? Participants were paid at a rate of \$0.05 per label, with an average rating time of 4 seconds per image-ending at an average rate of roughly \$45 hour.
 - Were participants given the ability to opt out? All submissions were voluntary.
 - Were participants told of the purpose of their work? Participants were told that their work was being used to "validate machine learning model performance".
 - Was any data or personal information collected from the participants? No personal data was collected from the participants.
- Was there any potential risks done to the participants? Although some ImageNet classes are sometimes known to contain elicit or unwelcome content Prabhu (2019). Our 100-class subset of ImageNet purposefully excludes such classes, and as such participants were not subject to any undue risks or personal harms.

Adversarial Images Classification

This work is used to validate machine learning model performance and your participation is voluntary. You're free to stop the task at any point in time. You'll be shown an image. One of the labels is indeed present in the image please select the correct one. If you're unfamiliar with a label take a second to search for it on google images. Please let us know if this happens often.

The image may however be too corrupted in which case select that it is too corrupted. Please avoid using corrupted label unless necessary.

Thanks!



Select which label is present in the image or if the image is too corrupted.

- Granny Smith (type of eating apple)
- pretzel (type of pretzel)
- pufferfish (type of fish)
- saxophone (type of musical instrument)
- accordion (type of musical instrument)
- Image is too corrupted

Next preview

Figure 23: Interface of participants. We demonstrate the interface which was provided to the participants of the study, involving the selection of correct classes from our 100-class subset of ImageNet.

```
1815
      This work is used to validate machine learning model performance and your
1816
      participation is voluntary. You're free to stop the task at any point in
1817
      time.
      You'll be shown an image. One of the labels is indeed present in the
1818
      image please select the correct one. If you're unfamiliar with a label
1819
      take a second to search for it on google images. Please let us know if
1820
      this happens often.
1821
      The image may however be too corrupted in which case select that it is
1822
      too corrupted. Please avoid using corrupted label unless necessary.
      Thanks!
1823
1824
      Figure 24: Instructions given to the participants. Above is a list of the instructions which were
1825
      given to the participants in the human study.
1826
1827
1828
1829
1830
1831
```



Figure 25: L_p robustness correlates with UA2. Across our benchmark, for adversarially trained models L_p robustness correlates with UA2 - however, several models trained without adversarial training still improve on UA2.

M GRID SEARCH VS. GRADIENT-BASED SEARCH

Table 18: Comparing gradient-based search to grid-based search We compare the performance of optimising with a randomised grid-based search using 1000 forward-passes per datapoint, finding that our gradient-based methods perform a lot better than this compute-intensive baseline.

Optimisation Technique	UA2
Randomized grid search	74.1
Gradient-based search (ours)	7.2

1	878
1	879
1	880

1890 N TRANSFER ATTACKS

1891

1896

1897

1912

1913

Table 19 shows the transfer-attack performances across various source and target models based on 1000 test samples. We observe that while the transfer attacks are not as effective as white-box attacks, they consistently outperform baseline unoptimized attacks where the perturbations are randomly initialized (Table 20).

Table 19: Transfer attack performance

	Clean Acc.	PGD	UA2	JPEG	Elastic	Wood	Glitch	Kal.	Pixel	Snow	Gabor
ResNet50 (source model)	75.2	0	13.2	0	22.2	30.8	10	4.3	4.8	3.1	30.4
ViT-small Patch16 ImageNet1K	78.5	73.1	59.99	75	62.7	69.9	46	48	62.8	55.5	60
ConvNeXt-V2-tiny ImageNet1K	82.1	74.8	67.66	77.1	69	75.9	54	60	73.6	65.2	66.5
Swin-small ImageNet1K + L_{∞} 4/255	71.1	70.6	50.39	70.9	56.7	65.8	34.8	10.7	59.3	48.4	56.5
ResNet50	75.2	67.9	43.19	70.1	53.1	57.7	30.1	5.4	53.3	38.1	37.7
ViT-small Patch16 ImageNet1K (source model)	78.5	0	6.51	0	8.2	12.7	0.5	4.7	2.1	0.8	23.1
ConvNeXt-V2-tiny ImageNet1K	82.1	75.7	67.3	78.6	68.5	72.8	56.4	59.9	70.1	65.1	67
Swin-small ImageNet1K + L_{∞} 4/255	71.1	70.5	50.11	70.9	57.1	65.1	35	10.8	59.5	48	54.5
ResNet50	75.2	67.8	42.06	68.3	51	55.7	31.7	5.8	51.7	32.1	40.2
ViT-small Patch16 ImageNet1K	78.5	74.7	57.31	75	60	69	42	46.8	57.2	50.2	58.3
ConvNeXt-V2-tiny ImageNet1K (source model)	82.1	0	12.15	0	23.2	22.3	7.4	3.5	6	0.6	34.2
Swin-small ImageNet1K + L_{∞} 4/255	71.1	71.2	50.1	71.2	56.1	65	37.8	10.7	59.1	45	55.9
ResNet50	75.2	64	36.95	61.8	42.5	57.8	15.6	5.4	45.3	29.2	38
ViT-small Patch16 ImageNet1K	78.5	66.9	53.3	70.6	51.4	68.2	23.8	47.1	58.4	44.2	62.7
ConvNeXt-V2-tiny ImageNet1K	82.1	75.5	65.26	75.7	64.7	74.5	46.1	58.2	72.3	63.6	67
Swin-small ImageNet1K + L_{∞} 4/255 (source model)	71.1	53.8	21.4	42	17.9	42.3	5.1	5.1	7.6	3.4	47.8

Table 20: Unoptimized attack performance

	Clean Acc.	PGD	UA2	JPEG	Elastic	Wood	Glitch	Kal.	Pixel	Snow	Gabor
ResNet50 ViT-small Patch16 ImageNet1K ConvNeXt-V2-tiny ImageNet1K Swin-small ImageNet1K + L_{∞} 4/255	75.2 78.5 82.1 71.1	74.1 78 82.2 71.3	56.44 69.19 74.74 58.19	74.3 78 82.2 71.6	62.8 70.2 75.2 62	55.7 70.2 74.4 63.4	55.8 65.4 69.7 58	6.3 47.7 60.7 10.2	74.1 77.3 81.5 70.9	74.8 78.6 81.4 71.7	47.7 66.1 72.8 57.7

1946 1947 1948 1949 194	1945														
1947 1948 1948 1949 194	1946														
1948 1949 1940 1941 1942 1944 1945 194	1947														
1949 194	1948														
Figure 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	1949														
Figure 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	1950														
Figure 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	1951														
Figure 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1	1952														
Fight for the second	1953														
Figure 26 (2012) 102 100 100 100 100 100 100 100 100 100	1954														
Figure 26: Figure 2	1955		Clean accPGD (4/255)	UA2	Edge	FBM	Fog	HSV	Klotski	Mix	Pokadot	Prison	Blur	Texture	Whirlpool
Figure 4.27. Fig	1956	VII-base Patch16 + L_{∞} 4/255 ConvNeXt-base + L_{∞} 4/255 Swip base imageNet1K + L_{∞} 4/255	- 76.8 - 57.1	45.6	- 23.2	63.0	59.1	51.4	27.4	- 64.8 - 64.5	- 12.4	- 42.5	65.2	32.0	- 50.6
Figure 26: Image: 1.272 Concentry transmission: 1.272 Concentry tran	1057	ConvNeXt-small + L ₌ 4/255 ViT-small Patch16 + L 4/255	74.1 - 54.4	39.8 39.3	22.8	59.2	52.8	48.7	26.6	61.5	- 6.5	27.2	59.2	28.0	45.4
1950 CLUMULAT 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1050	ConvNeXt-V2-large ImageNet1K+22K Swin-small ImageNet1K + L ₂ 4/255	- 87.3-0.0 - 73.4 - 50.7	37.4	-0.7	26.5	67.5	0.4 45.2	27.6	47.5	- 6.7	4 7.2	58.6	-0.0	- 73.4
1999 Control Market managements (22) 000	1950	CLIP (VIT-L/14) ConvNeXt-V2-base ImageNet1K+22K	- 75.5 -0.3	37.0	-2.1	35.8	62.9	7.1	- 53.0	47.9	56.0	30.7	49.9	-1.5	60.5
1990 Convectory impacted in the second of the second o	1959	ConvNeXt-v2-base ImageNet1K+22K ConvNeXt-xlarge ImageNet1K+22K	87.0-0.0	33.7	-0.5	- 18.7	62.7	0.1	- 57.6	41.1	64.0	-1.8	- 55.3	-0.0	68.7
1961 urbanker urb	1960	ConvNeXt-V2-huge ImageNet1K ConvNeXt-V2-huge ImageNet1K	- 86.3-0.0	32.5	-1.2	- 21.3	58.3	0.7	- 55.3	32.5	- 39.5	19.2	- 57.0	-0.3	- 72.6
1962 Convexts and imagenetic vizze 0.000 <	1961	ConvNeXt-large ImageNet1K+22K ViT-large Patch16 ImageNet1K+22K	- 85.8-0.1 ·	32.5	-0.5	- 16.2	60.8	0.6	- 39.1	39.5	44.0	27.9	46.0	-0.0	69.8
	1962	ConvNeXt-base ImageNet1K+22K ConvNeXt-V2-tiny ImageNet1K+22K	83.9-0.0	27.8	-0.3	10.8	57.6	0.2	45.7	31.7	52.3	-2.5	44.1	-0.0	60.9
1964 Convector superimediation a a a a a a a a	1963	ConvNeXt-V2-base ImageNet1K+22K ConvNeXt-small ImageNet1K+22K	- 84.9-0.0	27.7	-1.0	- 16.4	53.1 54.9	0.1	46.6	28.6	29.8	-12.3	49.6	-0.1	67.1
	1964	ConvNeXt-large ImageNet1K Swin-large ImageNet1K	- 84.3-0.0 86.3-0.0	26.8 26.0	-0.7 -0.3	12.1	44.7	0.2	53.6	24.6	- 16.3 - 29.8	23.5	- 54.5 - 43.7	-0.0 -0.0	64.3
Nexted by (1/2 are inside i	1965	ConvNeXt-base ImageNet1K Swin-base ImageNet1K	- 83.8-0.0 ·	26.0 25.9	-0.3 -0.1	- 10.9 - 15.5	45.8	0.0 1.1	51.8	26.3 23.4	- 25.4 - 32.3	- 12.0 - 9.7	- 50.3 41.5	-0.0 -0.0	62.8
Convexty Ando imageNet14-222 22.2 22.3 22.4 23.5 20.5 22.8 44.9 22.7	1966	Reversible-ViT-base Reversible-ViT-base multiscale	- 81.7-0.0 · · · · · · · · · · · · · · · · · ·	25.7 25.4	-1.1 -0.6	- 17.7 - 12.2	49.1 49.8	1.7 0.3	- 35.5 - 46.6	31.5 21.9	20.6 30.8	26.3 11.1	41.6 45.2	-0.0 -0.0	57.8 60.8
Convolution Convolution 883-10.0 223 10 10 223 10 10 223 10 10 223 10 10 223 10 10 223 10 10 223 10 10 220 10 220 10 10 220 10 10 220 10 10 220 10 10 220 10 10 220 10 10 220 10 10 10 220 10 10 10 10 10 10 10 10 <td>1967</td> <td>ConvNeXt-V2-nano ImageNet1K+22K ViT-base Patch16 ImageNet1K+22K</td> <td>- 82.0-0.0 ·</td> <td>25.4 25.0</td> <td>-0.4 -0.1</td> <td>- 9.9 - 8.3</td> <td>54.7 · 53.9 ·</td> <td>-0.0 -0.0</td> <td>40.4</td> <td>28.8</td> <td>44.9 34.9</td> <td>-3.8 - 15.2</td> <td>40.0 37.8</td> <td>-0.0 -0.0</td> <td>65.5</td>	1967	ConvNeXt-V2-nano ImageNet1K+22K ViT-base Patch16 ImageNet1K+22K	- 82.0-0.0 ·	25.4 25.0	-0.4 -0.1	- 9.9 - 8.3	54.7 · 53.9 ·	-0.0 -0.0	40.4	28.8	44.9 34.9	-3.8 - 15.2	40.0 37.8	-0.0 -0.0	65.5
1969 WT base factor is imageNetix 72.9 × 0 23.3 × 0.1 11.4 20.9 × 0.3 23.4 10.0 94.4 10.0 94.6 10.0 94.6 10.0 94.6 10.0 94.6 10.0 94.6 10.0 94.6 10.0 94.6 94.7 94.0 94.6 94.6 94.6 94.6 94.6 94.6 94.6 94.6 94.6 94.6 94.6 94.7 94.0 94.6 94.7 94.0 94.6 94.7 94.0 94.6 94.7 94.6 94.6 94.7 94.6 94.7 94.0 94.7 94.6 94.7 94.0 94.7 94.6 94.7 94.6 94.7	1068	ConvNeXt-small ImageNet1K ConvNeXt-tiny ImageNet1K+22K	- 83.1-0.0 · 82.9-0.0 ·	24.9 24.5	-0.5 -0.1	- 9.4 - 9.7	46.6	0.0 0.2	- 45.2	27.0	- 23.7 - 47.4	-2.4	49.6	-0.0 -0.0	- 59.4
1906 Redeted + L_3 / 25 -25 -017 -238 -241 -016 -216 -017 -016 -017 -016 -017 -0	1000	ViT-base Patch16 ImageNet1K ConvNeXt-V2-nano ImageNet1K	79.2 - 0.0 · · · · · · · · · · · · · · · · · ·	24.3	-0.1 -0.2	- 11.4 - 14.2	49.3	0.3 0.9	- 27.1 - 38.2 -	27.6	- 25.0	-16.0 -7.0	- 39.8 - 42.9	-0.0 -0.0	- 56.8 - 58.0
1970 Swin-small imageNetIX 332-10 233 -233 -233 -131 -107 -500 24 -530 -24 -500 -200 -600 -700 -600	1909	ResNet50 + L_2 3 ResNet50 + L_{∞} 4/255	62.8 - 31.7 · 63.9 - 39.0 ·	23.8	- 24.1 - 14.1	41.6	22.0	21.8	- 7.5	47.3 47.8	-0.6 -0.4	- 10.9 - 9.6	- 47.6 - 44.5	- 6.5 - 6.3	37.0
1971 Convexts try imageNetik B2.1 = 0.0 22.0 22.0 23.0 25.0 86.0 46.8 70.0 22.0 24.4 44.7 40.7 40.0 55.6 56.6 1972 Switch imageNetik Besters 0 + L, 2255 B0.1 B0.0 22.8 B0.2 B0.9 22.2 22.4 B0.9 22.6 B0.9 22.6 B0.9 22.6 B0.9 22.7 B0.0 B0.9 22.6 B0.9 22.7 B0.0 B0.9 22.6 B0.9 22.7 22.6 B0.9 22.	1970	Swin-small ImageNet1K ConvNeXt-V2-tiny ImageNet1K	- 83.2-0.0 - 82.9-0.0	23.5 23.2	-0.3 -0.1	- 13.7 - 10.7	37.9	0.4 0.4	- 53.0	24.6 21.5	- 17.6 - 20.7	-10.9 -15.6		-0.0 -0.0	60.7
1972 ResNet50 + L, 2/25 69.1 30.8 22.8 8.2 39.9 22.2 22.4 8.2 40.0 7.7 11.3 44.8 30.0 35.3 1973 ResNet50 + L, 1/25 80.3 10.0 22.6 3.3 10.5 50.5 50.5 50.5 7.0 10.8 42.8 40.0 42.8 40.0 55.6 50.5 7.0 10.3 40.8 40.3 40.0 42.8 40.0 55.6 50.5 7.0 10.5 40.0	1971	ConvNeXt-tiny ImageNet1K Reversible-ViT-small	- 82.1-0.0 ·	23.0 22.8	-0.5 -0.6	- 8.6 - 18.0	46.8	0.0 0.7	- 44.5 · 27.3 ·	24.0 29.6	-22.0 -16.4	-3.4 	-44.7 -36.1	-0.0 -0.1	- 58.5 - 56.6
1973 Convextvy-pice imageNet1K #0.3 + 0.1 #0.5 + 0.3 #0.5 + 0.1 #0.5 + 0.1 #0.6 + 0.2 + 0.1 #0.8 + 0.0 #0.0 + 0.0 <t< td=""><td>1972</td><td>ResNet50 + L_o 2/255 Swin-tiny ImageNet1K</td><td>- 69.1 - 30.8 · · · · · · · · · · · · · · · · · · ·</td><td>22.8</td><td>- 8.2 - 0.0</td><td>- 39.9 - 9.9</td><td>28.2</td><td>22.4 0.2</td><td>- 8.2</td><td>48.0 21.8</td><td>-0.7</td><td>-11.3 -4.7</td><td>-44.8 -37.3</td><td>-3.0 -0.0</td><td>- 36.3 - 59.6</td></t<>	1972	ResNet50 + L _o 2/255 Swin-tiny ImageNet1K	- 69.1 - 30.8 · · · · · · · · · · · · · · · · · · ·	22.8	- 8.2 - 0.0	- 39.9 - 9.9	28.2	22.4 0.2	- 8.2	48.0 21.8	-0.7	-11.3 -4.7	-44.8 -37.3	-3.0 -0.0	- 36.3 - 59.6
1974 ResNet50 - L 1255 172.0 112.1 22.4 -17.1 116.0 -14.8 -14.1 12.2 -16.6 -20.0 11.8 22.6 22.6 11.8 22.6 22.6 12.6 20.0 21.5 6.0 22.6 22.6 20.0 12.5 6.0 22.6 20.0 12.5 6.0 22.6 20.0 12.6 90.0 13.7 93.8 90.0 -11.7	1973	ConvNeXt-V2-pico ImageNet1K ResNet50 + L ₂ 5	- 80.3-0.0 - 56.1 - 34.1	22.6	-0.3	- 10.5 - 34.8	50.5	0.1	36.9	27.5	21.8	9.2	38.2	-0.0 - 10.3	- 53.9 - 34.8
1975 ResNet50 + L, 8/25 19,5 19,9 22,3 20,6 33,5 11,2 30,0 10,6 39,2 0,1 9,3 90,0 11,5 92,6 14,4 44,6 14,4 46,6 1,5 35,4 1976 ResNet50 + L_1 70,4 11,1 11,4 10,8 30,0 12,6 99,6 71,1 44,8 10,4 44,8 10,4 44,1 1,5 33,4 44,1 1,5 33,4 10,0 13,7 76,5 10,0 <	1974	ResNet50 + L_{∞} 1/255 ResNet50 + CutMix	72.0 - 18.1	22.4	-4.4 -0.7	- 37.3 - 16.0	- 34.8 46.5	4.4	- 8.2 - 43.3	46.6	-2.0 	- 15.8 - 6.0	- 44.3 - 26.6	-1.0 -0.1	38.0
No.100 Resetted > L_2 1 Tot 4 Tot 4 <td>1975</td> <td>ResNet50 + L_n 8/255 ResNet50 + Moex</td> <td>- 54.5 - 38.9 - 79.0 - 0.5</td> <td>22.3 21.9</td> <td>-0.8</td> <td>- 36.5 - 14.8</td> <td>43.5</td> <td>30.0 3.4</td> <td>- 10.6 - 43.9</td> <td>39.2 26.1</td> <td>-0.1 </td> <td>- 9.3 - 7.4</td> <td></td> <td>- 15.7 - 0.0</td> <td>- 32.8 - 53.4</td>	1975	ResNet50 + L _n 8/255 ResNet50 + Moex	- 54.5 - 38.9 - 79.0 - 0.5	22.3 21.9	-0.8	- 36.5 - 14.8	43.5	30.0 3.4	- 10.6 - 43.9	39.2 26.1	-0.1 	- 9.3 - 7.4		- 15.7 - 0.0	- 32.8 - 53.4
WTsmall Patch 16 ImageNet1k T78.8 0.0 T0.6 0.1 81.1 45.2 0.2 12.7 25.2 12.6 13.3 0.0 45.7 1977 UTsmall Patch 16 ImageNet1k T6.7 0.0 19.6 0.2 19.6 0.2 19.2 0.2 12.1 10.5 14.4 10.5 0.0 45.7 1978 ResNet50 + P.MMk T78.7 0.0 19.6 0.2 19.6 0.2 19.6 0.0 10.4 13.7 10.6 44.9 0.0 10.0	1076	ResNet50 + L ₂ 1 ResNet50 + L _x 0.5/255	- 70.4 - 15.1 · · · · · · · · · · · · · · · · · · ·	21.4	-10.8 -1.5	30.0	32.6 37.3	9.6 6.1	- 7.1 ·	45.8 44.9	-1.0 -2.2	-14.4 -15.4	46.1	-1.5 -0.3	
1977 ConvektV2-zato imageNetIX 76.7 ± 0.0 19.6 -0.2 19.2 -0.4 53.2 22.3 -1.4 -0.5 -0.6 -0.7 1978 ResNet50 + RvMik 77.5 ± 0.0 19.4 0.1 55.5 10.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.7 -0.6 -0.6 -0.7 -0.6 -0.7 -0.6 -0.6 -0.7 -0.6 -0.7 -0.6 -0.6 -0.6 -0.6 -0.6 -0.6 -0.6 -0.7 -0.6 -0.7 -0.6 -0.6 -0.6 -0.6 -0.7 -0.6 -0.7 -0.6 -0.6 -0.6 -0.6 -0.6 -0.6 -0.6 -0.7 -0.6	1077	ViT-small Patch16 ImageNet1K ViT-small Patch16 ImageNet1K+22K	- 78.8-0.0 · 81.4-0.0 ·	20.6 20.1	-0.1 -0.0	- 8.1 - 5.9	45.2 45.9	0.2 0.0	- 21.7	25.2 21.3	- 29.2 - 35.0	- 12.6 - 8.4	- 31.3 - 26.0	-0.0 -0.0	52.7 55.8
1976 Beshet50 + FMMk 78.1 - 0.0 19.4 -0.1 45.5 -0.0 12.9 19.8 -0.0 12.9 19.8 -0.0 12.9 19.8 -0.0 12.9 19.8 -0.0 12.7 10.8 -0.0 45.5 56.5 1979 ConvektV2-femto ImageNet1K 77.5 - 0.0 17.7 -0.4 66.7 -0.4 14.3 20.3 -10.7 10.8 -0.0 46.7 1980 ResNet50 + Mukup 77.5 - 0.0 17.7 -0.4 46.4 49.8 -0.0 10.8 20.3 -10.7 10.8 -0.0 46.7 1980 Peshet50 + Mukup 77.5 -0.0 11.7 -0.2 5.1 -0.0 10.8 20.4 49.0 10.8 20.3 -10.7 10.4 49.0 1981 ResNet50 + AugMik 77.5 -0.0 11.6.7 -0.1 5.1 -0.0 10.5 10.5 10.5 10.8 20.4 49.0 10.4 49.0 10.4 10.0 10.8 20.7 11.4 10.2 20.7 10.4 49.0 10.4 10.7 <t< td=""><td>1977</td><td>ConvNeXt-V2-atto ImageNet1K ResNet50 + L₂ 0.5</td><td>- 76.7 - 0.0 - 73.2 - 15.5</td><td>19.6 19.5</td><td>-0.2 -4.2</td><td>- 9.2 - 25.0</td><td>46.5 34.4</td><td>0.0 3.7</td><td>- 32.2</td><td>24.2 41.9</td><td>26.3 2.5</td><td>-1.4</td><td>- 31.5 - 42.9</td><td>-0.0 -0.5</td><td>44.7</td></t<>	1977	ConvNeXt-V2-atto ImageNet1K ResNet50 + L ₂ 0.5	- 76.7 - 0.0 - 73.2 - 15.5	19.6 19.5	-0.2 -4.2	- 9.2 - 25.0	46.5 34.4	0.0 3.7	- 32.2	24.2 41.9	26.3 2.5	-1.4	- 31.5 - 42.9	-0.0 -0.5	44.7
1979 Convextv2:femto imageNet1k 78.5 × 0.0 18.6 0.0 66.4 42.4 0.0 18.8 23.3 18.7 3.0 18.7 0.0 48.6 1980 ResNet50 + Nupp 75.8 × 0.0 17.2 0.2 5.1 47.1 0.0 18.6 22.2 12.3 18.7 10.0 18.6 0.0 48.6 1980 WT-base Pach23 ImageNet1k 77.5 × 0.0 17.2 0.2 5.1 47.1 0.0 10.2 22.0 12.7 18.8 0.0 48.6 1981 ResNet50 + AugMix 77.5 × 0.0 11.6.7 0.1 5.3 40.8 10.0 15.7 22.6 10.2 22.6 10.0 49.0 1981 ResNet50 + AugMix 77.5 × 0.0 11.6.7 11.5 41.1 10.0 10.5 10.2 18.6 10.4 49.0 1982 ResNet50 + RandAug 77.5 × 0.0 11.45 10.5 40.7 10.0 10.8 10.0 10.8 10.0 10.8 10.0 10.8 10.0 10.8 10.0 10.8 10.0 <td< td=""><td>1978</td><td>ResNet50 + PixMix ViT-base Patch32 ImageNet1K+22K</td><td>- 78.1 - 0.0 · · · · · · · · · · · · · · · · · ·</td><td>19.4</td><td>-0.1 -0.2</td><td>-15.5 -16.7</td><td>43.9</td><td>-0.0 -0.1</td><td>- 27.9</td><td>- 19.8 - 26.3</td><td>40.2</td><td>-1.7</td><td>- 18.3 - 31.3</td><td>-0.0 -0.0</td><td>- 56.0 - 54.5</td></td<>	1978	ResNet50 + PixMix ViT-base Patch32 ImageNet1K+22K	- 78.1 - 0.0 · · · · · · · · · · · · · · · · · ·	19.4	-0.1 -0.2	-15.5 -16.7	43.9	-0.0 -0.1	- 27.9	- 19.8 - 26.3	40.2	-1.7	- 18.3 - 31.3	-0.0 -0.0	- 56.0 - 54.5
1980 ResNet50 + DepAug+AugMix 7/3 + 0.0 10/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 1/2 9/2 9/2 9/2 1/2 9/2	1979	ConvNeXt-V2-femto ImageNet1K ResNet50 + Mixup	78.5 - 0.0	18.6 17.7	-0.0 -0.4	- 6.6 - 6.4	42.4	0.0	- 31.8	23.3	- 18.7 - 28.0	-3.0	30.7	-0.0	48.7
1981 1981 ResNet50 + Ly 0.1 WT.5mill Patch20 magNet1K+22K ResNet50 + ANT7/5 + 0.0 4 0.1 4 0.0 1001/5 + 0.1 4	1980	ResNet50 + DeepAug+AugMix ViT-base Patch32 ImageNet1K	74.9 - 0.0	17.2	-0.2	- 9.8	47.1	-0.0	- 9.6	25.7	- 12.4	-4./ -4.9	- 34.8	-0.0	41.2
1982 VI = small vsch32 mageNetIL+2x 70.0 + 0.0 + 0.1 + 0.0	1981	ResNet50 + AugMix ResNet50 + L ₂ 0.1	- 74.8 -0.1	15.9	-0.1	-10.5	44.1 · 39.8 ·	0.0	- 8.8	30.3	-4.7	-2.6	- 30.7 - 32.8	-0.0	- 37.5
1983 ResNet50 + National Stress 100^{-1} $100^$	1982	vii-small Patch32 ImageNet1K+22K ResNet50 + Deepaugment	76.7 -0.0	14.9	-0.2	-3.7	42.6	0.1	9.1	20.5	- 14.5	-2.6	23.5	-0.0	48.1
1984 Image: NetSO + ANT 100 + 100	1983	ResNet50 + RandAug ResNet50 + Stylised ImageNet	76.7 -0.0	13.2	-0.0	-1.9	39.7 ·	0.0	- 13.7	16.9	-14.3	-1.3	- 15.3	-0.0	41.3
1985 1986 1987 Figure 26: ImageNet UA2 performance under extra attacks in medium distortion	1984	vii-tiny Patchito imageNet1K+22K ResNet50	- 76.1 - 0.0	12.3	-0.0	-0.9	35.5	0.0	-13.6	13.7	- 15.2	-0.4	- 12.8	-0.0	43./
Figure 26: ImageNet UA2 performance under extra attacks in medium distortion	1985	NESNELOU + ANI	0 100 0 100	0 1	1 H	000 10	20 100	0 10	100 100	D0 10	00 10	00 10	100 10	00 10	000 100
Figure 26: ImageNet UA2 performance under extra attacks in medium distortion	1986	T '				C		1.	4	1		1.	1		
	1987	Figure	20: ImageN	et UA	12 per	iormar	ice un	der er	xtra att	acks 1	n meo	num d	uistort	ion	