TESTING ROBUSTNESS AGAINST UNFORESEEN ADVERSARIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Most existing defenses against adversarial attacks only consider robustness to L_p-bounded distortions. In reality, the specific attack is rarely known in advance and adversaries are free to modify images in ways which lie outside any fixed distortion model; for example, adversarial rotations lie outside the set of L_p-bounded distortions. In this work, we advocate measuring robustness against a much broader range of *unforeseen attacks*, attacks whose precise form is unknown during defense design.

We propose several new attacks and a methodology for evaluating a defense against a diverse range of unforeseen distortions. First, we construct novel adversarial JPEG, Fog, Gabor, and Snow distortions to simulate more diverse adversaries. We then introduce UAR, a summary metric that measures the robustness of a defense against a given distortion. Using UAR to assess robustness against existing and novel attacks, we perform an extensive study of adversarial robustness. We find that evaluation against existing L_p attacks yields redundant information which does not generalize to other attacks; we instead recommend evaluating against our significantly more diverse set of attacks. We further find that adversarial training against either one or multiple distortions fails to confer robustness to attacks with other distortion types. These results underscore the need to evaluate and study robustness against unforeseen distortions.

1 INTRODUCTION

Neural networks perform well on many benchmark tasks (He et al., 2016) yet can be fooled by adversarial examples (Goodfellow et al., 2014), slightly distorted inputs designed to subvert a given model. The adversary is frequently assumed to craft adversarial distortions under an L_∞ constraint (Goodfellow et al., 2014; Madry et al., 2017; Xie et al., 2018), while other distortions such as adversarial geometric transformations, patches, and even 3D-printed objects have also been considered (Engstrom et al., 2017; Brown et al., 2017; Athalye et al., 2017). However, most work on adversarial robustness assumes the adversary is *fixed* and known. Defenses against adversarial attacks often leverage such knowledge when designing the defense, most commonly through adversarial training, which minimizes the adversarial loss against a fixed distortion type (Madry et al., 2017).

In practice, adversaries can modify their attacks and construct distortions whose precise form is not known to the defense designers. In this work, we propose a methodology for assessing robustness to such *unforeseen attacks* and use it to study how adversarial robustness transfers to them. To ensure sufficient diversity, we introduce four novel adversarial attacks (§2) with qualitatively different distortion types: adversarial JPEG, Fog, Gabor, and Snow (sample images in Figure 1).

Our methodology (§3) involves evaluating a defense against a diverse set of held-out distortions not involved in the design of the defense; we suggest L_∞, L_1, Elastic, Fog, and Snow as an initial set to consider. For a fixed, held-out distortion, we then evaluate the defense against the distortion for a calibrated range of distortion sizes whose strength is roughly comparable across distortions. For each fixed distortion, our evaluation yields the summary metric UAR, which measures robustness of a defense against that distortion relative to a model adversarially trained on that distortion. We provide code and calibrations to easily evaluate a defense against our suite of attacks and compute UAR for it at `https://github.com/iclr-2020-submission/advex-uar`.

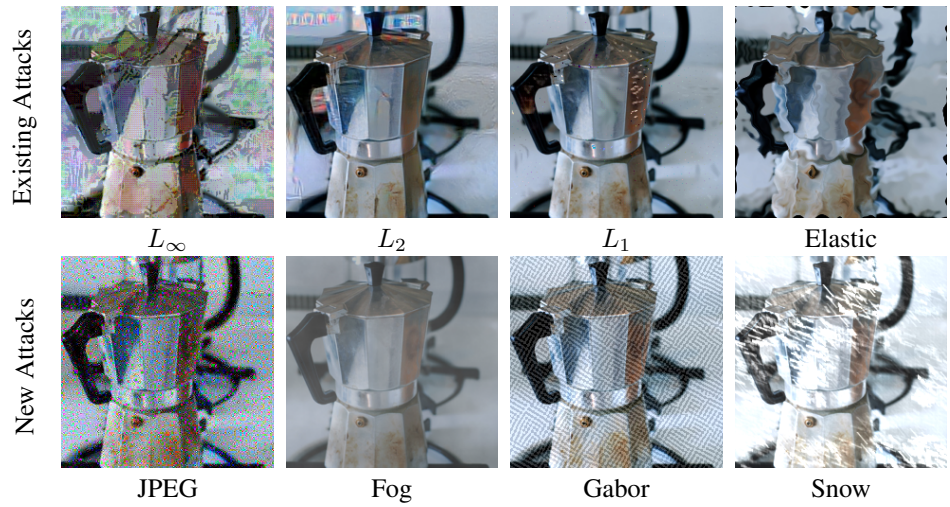


Figure 1: Attacked images (label "espresso maker") against adversarially trained models with large ε. Each of the adversarial images above are optimized to maximize the classification loss.

Applying our method to 87 adversarially trained models and 8 different distortion types (§4), we find weaknesses in existing defenses and evaluation practices. Our results show that existing defenses based on adversarial training do not generalize to unforeseen adversaries, even when restricted to the 8 distortions in Figure 1. This adds to the mounting evidence that achieving robustness against a single distortion type is insufficient to impart robustness to unforeseen attacks (Jacobsen et al., 2019; Jordan et al., 2019; Tramèr & Boneh, 2019).

Turning to evaluation, our results demonstrate that accuracy against different L_p distortions is highly correlated relative to the other distortions we consider, suggesting that the common practice of evaluating only against L_p distortions can give a misleading account of a model's adversarial robustness. Our analysis using UAR demonstrates that our full suite of attacks adds signficant diversity and reveals L_∞, L_1, Elastic, Fog, and Snow as a set with less correlated accuracy and UAR scores against held-out defenses. We suggest these attacks for use when evaluating against unforeseen adversaries.

A natural next approach is to defend against multiple distortion types simultaneously in the hope that seeing a larger space of distortions provides greater transfer to unforeseen distortions. Unfortunately, we find that defending against even two different distortion types via joint adversarial training is difficult (§5). Specifically, joint adversarial training leads to overfitting at moderate distortion sizes.

In summary, we make the following contributions:

1. We propose a method UAR to assess robustness of defenses against unforeseen adversaries.
2. We introduce 4 novel attacks and apply UAR to assess how robustness transfers to these attacks and 4 existing ones. Our results demonstrate that existing defense and evaluation methods do not generalize well to unforeseen attacks.
3. We suggest the use of our more diverse attacks for evaluating novel defenses, highlighting L_∞, L_1, Elastic, Fog, and Snow as a diverse starting point.

2 A SET OF DIVERSE AND NOVEL ADVERSARIAL ATTACKS

We consider distortions (attacks) applied to an image $x \in \mathbb{R}^{3\times224\times224}$, represented as a vector of RGB values. Let $f : \mathbb{R}^{3\times224\times224} \to \mathbb{R}^{100}$ be a model mapping images to logits[1], and let $\ell(f(x), y)$ denote the cross-entropy loss. For an input x with true label y and a target class $y' \neq y$, our adversarial attacks attempt to find x' such that

1. the attacked image x' is obtained by applying a constrained distortion to x, and
2. the loss $\ell(f(x'), y')$ is minimized (targeted attack).

[1]We describe the attacks for ImageNet-100, but they can also be applied to CIFAR-10.

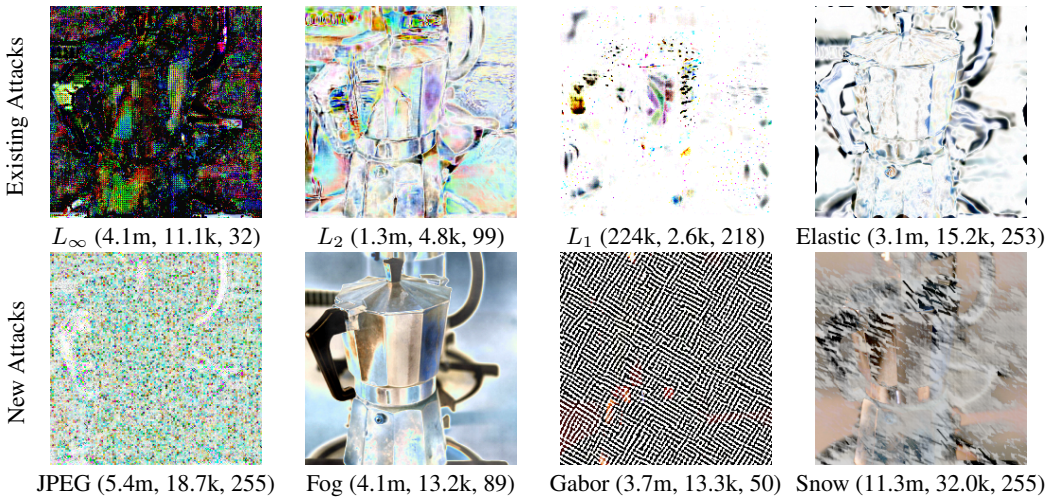


Figure 2: Scaled pixel-level differences between original and attacked images for each attack (label "espresso maker"). The L_1, L_2, and L_∞ norms of the difference are shown after the attack name. Our novel attacks display behavior which is qualitatively different from that of the L_p attacks. Attacked images are shown in Figure 1, and unscaled differences are shown in Figure 9, Appendix B.1.

Adversarial training (Goodfellow et al., 2014) is a strong defense baseline against a fixed attack (Madry et al., 2017; Xie et al., 2018) which updates using an attacked image x' instead of the clean image x at each training iteration.

We consider 8 attacks: L_∞ (Goodfellow et al., 2014), L_2 (Szegedy et al., 2013; Carlini & Wagner, 2017), L_1 (Chen et al., 2018), Elastic (Xiao et al., 2018), JPEG, Fog, Gabor, and Snow. We show sample attacked images in Figure 1 and the corresponding distortions in Figure 2. The JPEG, Fog, Gabor, and Snow attacks are new to this paper, and the L_1 attack uses the Frank-Wolfe algorithm to improve on previous L_1 attacks. We now describe the attacks, whose distortion sizes are controlled by a parameter ε. We clamp output pixel values to $[0, 255]$.

Existing attacks. The L_p attacks with $p \in \{1, 2, \infty\}$ modify an image x to an attacked image $x' = x+\delta$. We optimize δ under the constraint $\|\delta\|_p \leq \varepsilon$, where $\|\cdot\|_p$ is the L_p-norm on $\mathbb{R}^{3\times224\times224}$.

The Elastic attack warps the image by allowing distortions $x' = \mathsf{Flow}(x, V)$, where $V : \{1, \ldots, 224\}^2 \to \mathbb{R}^2$ is a vector field on pixel space, and Flow sets the value of pixel (i, j) to the bilinearly interpolated original value at $(i, j) + V(i, j)$. We construct V by smoothing a vector field W by a Gaussian kernel (size 25×25, std. dev. 3 for a 224×224 image) and optimize W under $\|W(i, j)\|_\infty \leq \varepsilon$ for all i, j. This differs in details from Xiao et al. (2018) but is similar in spirit.

Novel attacks. As discussed in Shin & Song (2017) for defense, JPEG compression applies a lossy linear transformation JPEG based on the discrete cosine transform to image space, followed by quantization. The JPEG attack imposes the L_∞-constraint $\|\mathsf{JPEG}(x) - \mathsf{JPEG}(x')\|_\infty \leq \varepsilon$ on the attacked image x'. We optimize $z = \mathsf{JPEG}(x')$ and apply a right inverse of JPEG to obtain x'.

Our novel Fog, Gabor, and Snow attacks are adversarial versions of non-adversarial distortions proposed in the literature. Fog and Snow introduce adversarially chosen partial occlusions of the image resembling the effect of mist and snowflakes, respectively; stochastic versions of Fog and Snow appeared in Hendrycks & Dietterich (2019). Gabor superimposes adversarially chosen additive Gabor noise (Lagae et al., 2009) onto the image; a stochastic version appeared in Co et al. (2019). These attacks work by optimizing a set of parameters controlling the distortion

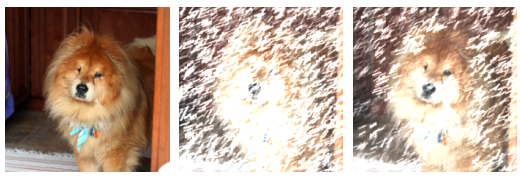


Figure 3: Snow before and after optimization.

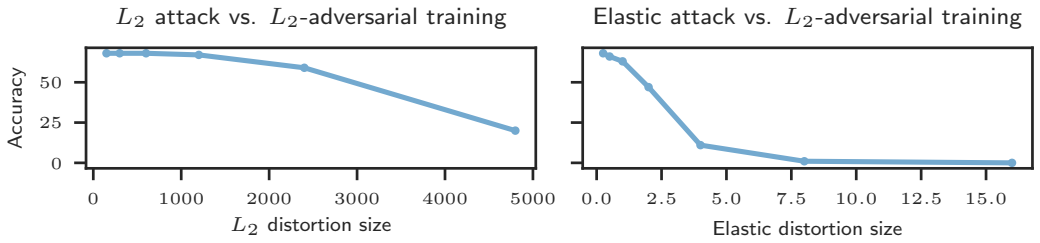


Figure 4: Accuracies of L_2 and Elastic attacks at different distortion sizes against a ResNet-50 model adversarially trained against L_2 at $\varepsilon = 9600$ on ImageNet-100. At small distortion sizes, the model appears to defend well against Elastic, but large distortion sizes reveal a lack of transfer.

over an L_∞-bounded set. Specifically, values for the diamond-square algorithm, sparse noise, and snowflake brightness (Figure 3) are chosen adversarially for Fog, Gabor, and Snow, respectively.

Optimization. To handle L_∞ and L_2 constraints, we use randomly-initialized projected gradient descent (PGD), which optimizes the distortion δ by gradient descent and projection to the L_∞ and L_2 balls (Madry et al., 2017). For L_1 constraints, this projection is more difficult, and previous L_1 attacks resort to heuristics (Chen et al., 2018; Tramèr & Boneh, 2019). We use the randomly-initialized Frank-Wolfe algorithm (Frank & Wolfe, 1956), which replaces projection by a simpler optimization of a linear function at each step (pseudocode in Appendix B.2).

3 Motivation and Description of Our Methodology

We now propose a method to assess robustness against unforeseen distortions, which relies on evaluating a defense against a diverse set of attacks that were *not* used when designing the defense. Our method must address the following issues:

- The range of distortion sizes must be wide enough to avoid the misleading behavior in which robustness appears to transfer at low distortion sizes but not at high distortion sizes (Figure 4);
- The set of attacks considered must be sufficiently diverse.

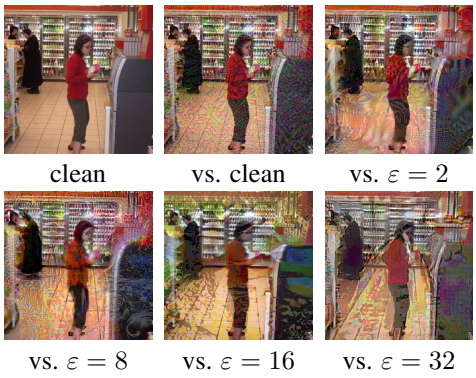
We first provide a method to calibrate distortion sizes and then use it to define a summary metric that assesses the robustness of a defense against a specific unforeseen attack. Using this metric, we are able to assess diversity and recommend a set of attacks to evaluate against.

Calibrate distortion size using adversarial training. As shown in Figure 4, the correlation between adversarial robustness against different distortion types may look different for different ranges of distortion sizes. It is therefore critical to evaluate on a wide enough range of distortion size ε. We choose the minimum and maximum distortion sizes ε using the following principles; sample images at ε_{min} and ε_{max} are shown in Figure 5b.

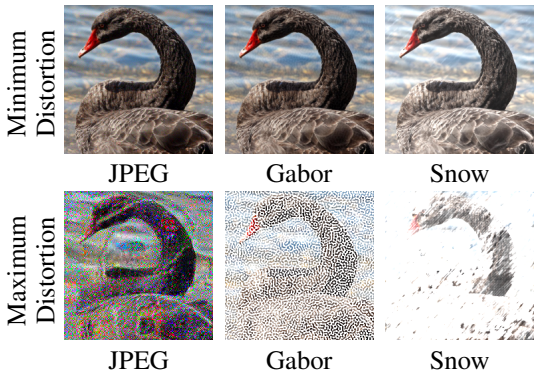
1. The minimum distortion size ε_{min} is the largest ε for which the adversarial validation accuracy against an adversarially trained model is comparable to that of a model trained and evaluated on unattacked data.
2. The maximum distortion size ε_{max} is the smallest ε which either (a) yields images which confuse humans when applied against adversarially trained models or (b) reduces accuracy of adversarially trained models to below 25.

In practice, we select ε_{min} and ε_{max} according to these criteria from a sequence of ε which is geometrically increasing with ratio 2. We choose to evaluate against adversarially trained models because attacking against strong defenses is necessary to produce strong visual distortions (Figure 5a). We introduce the constraint that humans recognize attacked images at ε_{max} because we find cases for L_1, Fog, and Snow where adversarially trained models maintain non-zero accuracy for distortion sizes producing images incomprehensible to humans. An example for Snow is shown in Figure 5b.

UAR**: an adversarial robustness metric.** We measure a model's robustness against a specific distortion type by comparing it to adversarially trained models, which represent an approximate



(a) The L_∞ attack at $\varepsilon = 32$ applied to undefended model and models adversarially trained against L_∞ at different distortion sizes. Attacking models trained against larger ε produces greater visual distortion.



(b) The JPEG, Gabor, and Snow attacks applied to adversarially trained models at ε_{min} and ε_{max}. Distortions are almost imperceptible at ε_{min}, but make the image barely recognizable by humans at ε_{max}.

Figure 5: Varying distortion size against adversarially trained models reveals full attack strength.

ceiling on performance with prior knowledge of the distortion type. For distortion type A and size ε, let the **A**dversarial **T**raining **A**ccuracy $\mathsf{ATA}(A, \varepsilon)$ be the best adversarial accuracy on the validation set that can be achieved by adversarially training a specific architecture (ResNet-50 for ImageNet-100, ResNet-56 for CIFAR-10) against A.[2] Even when evaluating a defense using an architecture other than ResNet-50 or ResNet-56, we recommend using the ATA values computed with these architectures to allow for uniform comparisons.

Given a set of distortion sizes $\{\varepsilon_1, \ldots, \varepsilon_n\}$, we propose the summary metric UAR (*Unforeseen Attack Robustness*) normalizing the accuracy of a model M against adversarial training accuracy:

$$\mathsf{UAR}(A, M) := 100 \cdot \left(\frac{1}{n} \sum_{k=1}^{n} \mathsf{Acc}(A, \varepsilon_k, M) \right) \Bigg/ \left(\frac{1}{n} \sum_{k=1}^{n} \mathsf{ATA}(A, \varepsilon_k) \right). \tag{1}$$

Here $\mathsf{Acc}(A, \varepsilon, M)$ is the accuracy of M against distortions of type A and magnitude ε. We expect most UAR scores to be lower than 100 against held-out distortion types, as an UAR score greater than 100 means that a defense is outperforming an adversarially trained model on that distortion. The normalizing factor in (1) is required to keep UAR scores roughly comparable between distortions, as different distortions can have different strengths as measured by ATA at the chosen distortion sizes.

Having too many or too few ε_k values in a certain range may cause an attack to appear artificially strong or weak because the functional relation between distortion size and attack strength (measured by ATA) varies between attacks. To make UAR roughly comparable between distortions, we evaluate at ε increasing geometrically from ε_{min} to ε_{max} by factors of 2 and take the subset of ε whose ATA values have minimum ℓ_1-distance to the ATA values of the L_∞ attack at geometrically increasing ε.

For our 8 distortion types, we provide reference values of $\mathsf{ATA}(A, \varepsilon)$ on this calibrated range of 6 distortion sizes on ImageNet-100 (Table 1, §4) and CIFAR-10 (Table 3, Appendix C.3.2). This allows UAR computation for a new defense using 6 adversarial evaluations and no adversarial training, reducing computational cost from 192+ to 6 NVIDIA V100 GPU-hours on ImageNet-100.

Evaluate against diverse distortion types. Since robustness against different distortion types may have low or no correlation (Figure 6b), measuring performance on different distortions is important to avoid overfitting to a specific type, especially when a defense is constructed with it in mind (as with adversarial training). Our results in §4 demonstrate that choosing appropriate distortion types to evaluate against requires some care, as distortions such as L_1, L_2, and L_∞ that may seem different can actually have highly correlated scores against defenses (see Figure 6). We instead recommend evaluation against our more diverse attacks, taking the L_∞, L_1, Elastic, Fog, and Snow attacks as a starting point.

[2] As explained in Figure 13 (Appendix C.2), this usually requires training at distortion size $\varepsilon' > \varepsilon$ because the typical distortion seen during adversarial training is sub-maximal.

Table 1: Calibrated distortion sizes and ATA values for different distortion types on ImageNet-100. ATA values for CIFAR-10 are shown in Table 3 (Appendix C.3.2).

Attack	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ATA_1	ATA_2	ATA_3	ATA_4	ATA_5	ATA_6
L_∞	1	2	4	8	16	32	84.6	82.1	76.2	66.9	40.1	12.9
L_2	150	300	600	1200	2400	4800	85.0	83.5	79.6	72.6	59.1	19.9
L_1	9562.5	19125	76500	153000	306000	612000	84.4	82.7	76.3	68.9	56.4	36.1
Elastic	0.250	0.500	2	4	8	16	85.9	83.2	78.1	75.6	57.0	22.5
JPEG	0.062	0.125	0.250	0.500	1	2	85.0	83.2	79.3	72.8	34.8	1.1
Fog	128	256	512	2048	4096	8192	85.8	83.8	79.0	68.4	67.9	64.7
Snow	0.062	0.125	0.250	2	4	8	84.0	81.1	77.7	65.6	59.5	41.2
Gabor	6.250	12.500	25	400	800	1600	84.0	79.8	79.8	66.2	44.7	14.6

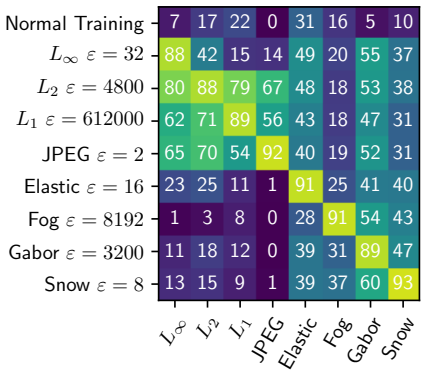
4 UAR REVEALS THE NEED TO EVALUATE AGAINST MORE DIVERSE ATTACKS

We apply our methodology to the 8 attacks in §2 using models adversarially trained against these attacks. Our results reveal that evaluating against the commonly used L_p-attacks gives highly correlated information which does not generalize to other unforeseen attacks. Instead, they suggest that evaluating on diverse attacks is necessary and identify a set of 5 attacks with low pairwise robustness transfer which we suggest as a starting point when assessing robustness to unforeseen adversaries.

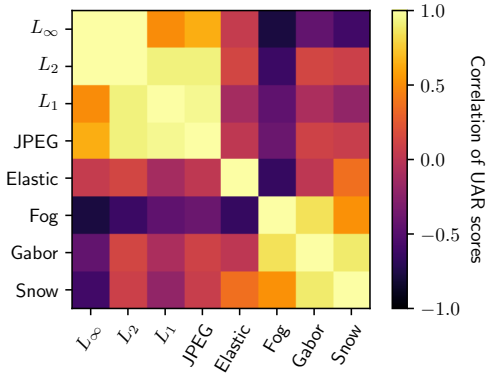
Dataset and model. We use two datasets: CIFAR-10 and ImageNet-100, the 100-class subset of ImageNet-1K (Deng et al., 2009) containing every 10^{th} class by WordNet ID order. We use ResNet-56 for CIFAR-10 and ResNet-50 as implemented in `torchvision` for ImageNet-100 (He et al., 2016). We give training hyperparameters in Appendix A.

Adversarial training and evaluation procedure. We construct hardened models using adversarial training (Madry et al., 2017). To train against attack A, for each mini-batch of training images, we select a uniform random (incorrect) target class for each image. For maximum distortion size ε, we apply the targeted attack A to the current model with distortion size $\varepsilon' \sim \text{Uniform}(0, \varepsilon)$ and update the model with a step of stochastic gradient descent using only the resulting adversarial images (no clean images). The random size scaling improves performance especially against smaller distortions. We use 10 optimization steps for all attacks during training except for Elastic, where we use 30 steps due to its more difficult optimization problem. When PGD is used, we use step size $\varepsilon/\sqrt{\text{steps}}$, the optimal scaling for non-smooth convex functions (Nemirovski & Yudin, 1978; 1983).

We adversarially train 87 models against the 8 attacks from §2 at the distortion sizes described in §3 and evaluate them on the ImageNet-100 and CIFAR-10 validation sets against 200-step targeted attacks with uniform random (incorrect) target class. This uses more steps for evaluation than train-



(a) UAR scores for adv. trained defenses (rows) against attacks (columns) on ImageNet-100. See Figure 12 for more ε values and Appendix C.3.2 for CIFAR-10 results.



(b) Correlations between UAR scores in Figure 6a for each attack (rows and columns). Correlation was computed over adversarial defenses in Figure 6a trained without knowledge of the attacks (6 total per pair).

Figure 6: UAR scores demonstrate the need to evaluate against diverse attacks.

ing per best practices (Carlini et al., 2019). We use UAR to analyze the results in the remainder of this section, directing the reader to Figures 10 and 11 (Appendix C.2) for exhaustive results and to Appendix D for checks for robustness to random seed and number of attack steps.

Existing defense and evaluation methods do not generalize to unforeseen attacks. The many low off-diagonal UAR scores in Figure 6a make clear that while adversarial training is a strong baseline against a fixed distortion, it only rarely confers robustness to unforeseen distortions. Notably, we were not able to achieve a high UAR against Fog except by directly adversarially training against it. Despite the general lack of transfer in Figure 6a, the fairly strong transfer between the L_p-attacks is consistent with recent progress in simultaneous robustness to them (Croce & Hein, 2019).

Figure 6b shows correlations between UAR scores of pairs of attacks A and A' against defenses adversarially trained without knowledge[3] of A or A'. The results demonstrate that defenses trained without knowledge of L_p-attacks have highly correlated UAR scores against the different L_p attacks, but this correlation does not extend to their evaluations against other attacks. This suggests that L_p-evaluations offer limited diversity and may not generalize to other unforeseen attacks.

The L_∞, L_1, Elastic, Fog, and Snow attacks offer greater diversity. Our results on L_p-evaluation suggest that more diverse attack evaluation is necessary for generalization to unforeseen attacks. As the unexpected correlation between UAR scores against the pairs $(\text{Fog}, \text{Gabor})$ and (JPEG, L_1) in Figure 6b demonstrates, even attacks with very different distortions may have correlated behaviors. Considering all attacks in Figure 6 together results in signficantly more diversity, which we suggest for evaluation against unforeseen attacks. We suggest the 5 attacks (L_∞, L_1, Elastic, Fog, and Snow) with low UAR against each other and low correlation between UAR scores as a good starting point.

5 Joint adversarial training: defending against two distortions

A natural idea to improve robustness against unforeseen adversaries is to adversarially train the same model against two different types of distortions simultaneously, with the idea that this will cover a larger portion of the space of distortions. We refer to this as *joint adversarial training* (Jordan et al., 2019; Tramèr & Boneh, 2019). For two attacks A and A', at each training step, we compute the attacked image under both A and A' and backpropagate with respect to gradients induced by the image with greater loss. This corresponds to the "max" loss described in Tramèr & Boneh (2019). We jointly train models for (L_∞, L_2), (L_∞, L_1), and $(L_\infty, \text{Elastic})$ using the same setup as before

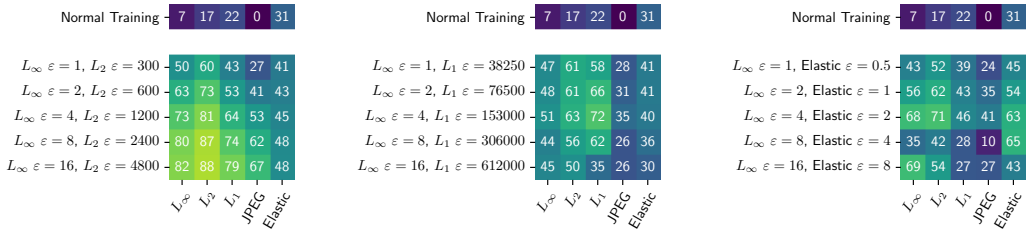


Figure 7: UAR scores for jointly adv. trained defenses (rows) against distortion types (columns).

Transfer for jointly trained models. Figure 7 reports UAR scores for jointly trained models using ResNet-50 on ImageNet-100; full evaluation accuracies are in Figure 19 (Appendix E). Comparing to Figure 6a and Figure 12 (Appendix E), we see that, relative to training against only L_2, joint training against (L_∞, L_2) slightly improves robustness against L_1 without harming robustness against other attacks. In contrast, training against (L_∞, L_1) is worse than *either* training against L_1 or L_∞ separately (except at small ε for L_1). Training against $(L_\infty, \text{Elastic})$ also performs poorly.

Joint training and overfitting. Jointly trained models achieve high *training* accuracy but poor validation accuracy (Figure 8) that fluctuates substantially for different random seeds (Table 4, Appendix E.2). Figure 8 shows the overfitting behavior for $(L_\infty, \text{Elastic})$: L_∞ validation accuracy decreases significantly during training while training accuracy increases. This contrasts with standard adversarial training (Figure 8), where validation accuracy levels off as training accuracy increases.

[3]We exclude defenses adversarially trained against A and A' to ensure that attacks are unforeseen.

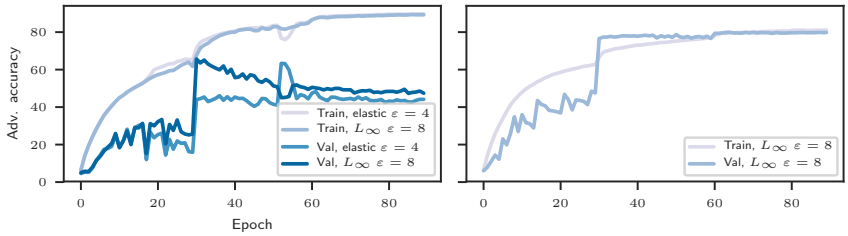


Figure 8: **Left:** train and validation curves for joint training against L_∞, $\varepsilon = 8$ and Elastic, $\varepsilon = 4$, **Right:** train and val curves for standard adversarial training for L_∞, $\varepsilon = 8$. The joint validation accuracy of L_∞ decreases as training progresses, indicating overfitting.

Overfitting primarily occurs when training against large distortions. We successfully trained against the (L_∞, L_1) and $(L_\infty, \text{Elastic})$ pairs for small distortion sizes with accuracies comparable to but slightly lower than observed in Figure 11 for training against each attack individually (Figure 18, Appendix E). This agrees with behavior reported by Tramèr & Boneh (2019) on CIFAR-10. Our intuition is that harder training tasks (more diverse distortion types, larger ε) make overfitting more likely. We briefly investigate the relation between overfitting and model capacity in Appendix E.3; validation accuracy appears slightly increased for ResNet-101, but overfitting remains.

6 Discussion and related work

We have seen that robustness to one attack provides limited information about robustness to other attacks, and moreover that adversarial training provides limited robustness to unforeseen attacks. These results suggest a need to modify or move beyond adversarial training. While joint adversarial training is one possible alternative, our results show it often leads to overfitting. Even ignoring this, it is not clear that joint training would confer robustness to attacks outside of those trained against.

Evaluating robustness has proven difficult, necessitating detailed study of best practices even for a single fixed attack (Papernot et al., 2017; Athalye et al., 2018). We build on these best practices by showing how to choose and calibrate a diverse set of unforeseen attacks. Our work is a supplement to existing practices, not a replacement–we strongly recommend following the guidelines in (Papernot et al., 2017) and (Athalye et al., 2018) in addition to our recommendations.

Some caution is necessary when interpreting specific numeric results in our paper. Many previous implementations of adversarial training fell prone to gradient masking (Papernot et al., 2017; Engstrom et al., 2018), with apparently successful training occurring only recently (Madry et al., 2017; Xie et al., 2018). While evaluating with moderately many PGD steps (200) helps guard against this, (Qian & Wegman, 2019) shows that an L_∞-trained model that appeared robust against L_2 actually had substantially less robustness when evaluating with 10^6 PGD steps. If this effect is pervasive, then there may be even less transfer between attacks than our current results suggest.

For evaluating against a fixed attack, DeepFool Moosavi-Dezfooli et al. (2015) and CLEVER Weng et al. (2018) can be seen as existing alternatives to UAR. They work by estimating "empirical robustness", which is the expected minimum ε needed to successfully attack an image. However, these apply only to attacks which optimize over an L_p-ball of radius ε, and CLEVER can be susceptible to gradient masking Goodfellow (2018). In addition, empirical robustness is equivalent to linearly averaging accuracy over ε, which has smaller dynamic range than the geometric average in UAR.

Our results add to a growing line of evidence that evaluating against a single known attack type provides a misleading picture of the robustness of a model (Sharma & Chen, 2017; Engstrom et al., 2017; Jordan et al., 2019; Tramèr & Boneh, 2019; Jacobsen et al., 2019). Going one step further, we believe that robustness itself provides only a narrow window into model behavior; in addition to robustness, we should seek to build a diverse toolbox for understanding machine learning models, including visualization (Olah et al., 2018; Zhang & Zhu, 2019), disentanglement of relevant features (Geirhos et al., 2018), and measurement of extrapolation to different datasets (Torralba & Efros, 2011) or the long tail of natural but unusual inputs (Hendrycks et al., 2019). Together, these windows into model behavior can give us a clearer picture of how to make models reliable in the real world.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *CoRR*, abs/1707.07397, 2017. URL http://arxiv.org/abs/1707.07397.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 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.

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

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian J. Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *CoRR*, abs/1902.06705, 2019. URL http://arxiv.org/abs/1902.06705.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. EAD: Elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Kenneth T. Co, Luis Muñoz-González, and Emil C. Lupu. Sensitivity of deep convolutional networks to Gabor noise. *CoRR*, abs/1906.03455, 2019. URL http://arxiv.org/abs/1906.03455.

Francesco Croce and Matthias Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. *CoRR*, abs/1905.11213, 2019. URL http://arxiv.org/abs/1905.11213.

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.

Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018. URL http://arxiv.org/abs/1811.12231.

Ian Goodfellow. Gradient masking causes CLEVER to overestimate adversarial perturbation size. *arXiv preprint arXiv:1804.07870*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Jrn-Henrik Jacobsen, Jens Behrmannn, Nicholas Carlini, Florian Tramr, and Nicolas Papernot. Exploiting excessive invariance caused by norm-bounded adversarial robustness, 2019.

Matt Jordan, Naren Manoj, Surbhi Goel, and Alexandros G. Dimakis. Quantifying perceptual distortion of adversarial examples. *arXiv e-prints*, art. arXiv:1902.08265, Feb 2019.

Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse Gabor convolution. *ACM Trans. Graph.*, 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`.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. DeepFool: a simple and accurate method to fool deep neural networks. *arXiv preprint arXiv:1511.04599*, 2015.

Arkadi Nemirovski and D Yudin. On Cezari's convergence of the steepest descent method for approximating saddle point of convex-concave functions. In *Soviet Math. Dokl*, volume 19, pp. 258–269, 1978.

Arkadi Nemirovski and D Yudin. *Problem Complexity and Method Efficiency in Optimization*. Intersci. Ser. Discrete Math. Wiley, New York, 1983.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519. ACM, 2017.

Haifeng Qian and Mark N. Wegman. L_2-nonexpansive neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=ByxGSsR9FQ`.

Yash Sharma and Pin-Yu Chen. Attacking the Madry defense model with L_1-based adversarial examples. *arXiv e-prints*, art. arXiv:1710.10733, Oct 2017.

Richard Shin and Dawn Song. JPEG-resistant adversarial images. In *NIPS 2017 Workshop on Machine Learning and Computer Security*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. *arXiv e-prints*, art. arXiv:1904.13000, Apr 2019.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018.

Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2019.

A TRAINING HYPERPARAMETERS

For ImageNet-100, we trained on machines with 8 NVIDIA V100 GPUs using standard data augmentation He et al. (2016). Following best practices for multi-GPU training Goyal et al. (2017), we ran synchronized SGD for 90 epochs with batch size 32×8 and a learning rate schedule with 5 "warm-up" epochs and a decay at epochs 30, 60, and 80 by a factor of 10. Initial learning rate after warm-up was 0.1, momentum was 0.9, and weight decay was 10^{-4}. For CIFAR-10, we trained on a single NVIDIA V100 GPU for 200 epochs with batch size 32, initial learning rate 0.1, momentum 0.9, and weight decay 10^{-4}. We decayed the learning rate at epochs 100 and 150.

B FURTHER ATTACK DETAILS

B.1 FURTHER EXAMPLES OF ATTACKS

We show the images corresponding to the ones in Figure 2, with the exception that they are not scaled. The non-scaled images are shown in Figure 9.

B.2 L_1 ATTACK

We chose to use the Frank-Wolfe algorithm for optimizing the L_1 attack, as Projected Gradient Descent would require projecting onto a truncated L_1 ball, which is a complicated operation. In contrast, Frank-Wolfe only requires optimizing linear functions $g^\top x$ over a truncated L_1 ball; this can be done by sorting coordinates by the magnitude of g and moving the top k coordinates to the boundary of their range (with k chosen by binary search). This is detailed in Algorithm 1.

C FULL EVALUATION RESULTS

C.1 L_1-JPEG AND L_2-JPEG ATTACKS

We will present results with two additional versions of the JPEG attack which impose L_1 or L_2 constraints on the attack in JPEG-space instead of the L_∞ constraint discussed in Section 2. To avoid confusion, in this appendix, we denote the original JPEG attack by L_∞-JPEG and these variants by L_1-JPEG and L_2-JPEG, respectively. Comparing the L_1-JPEG and L_2-JPEG attacks in Figure 10,

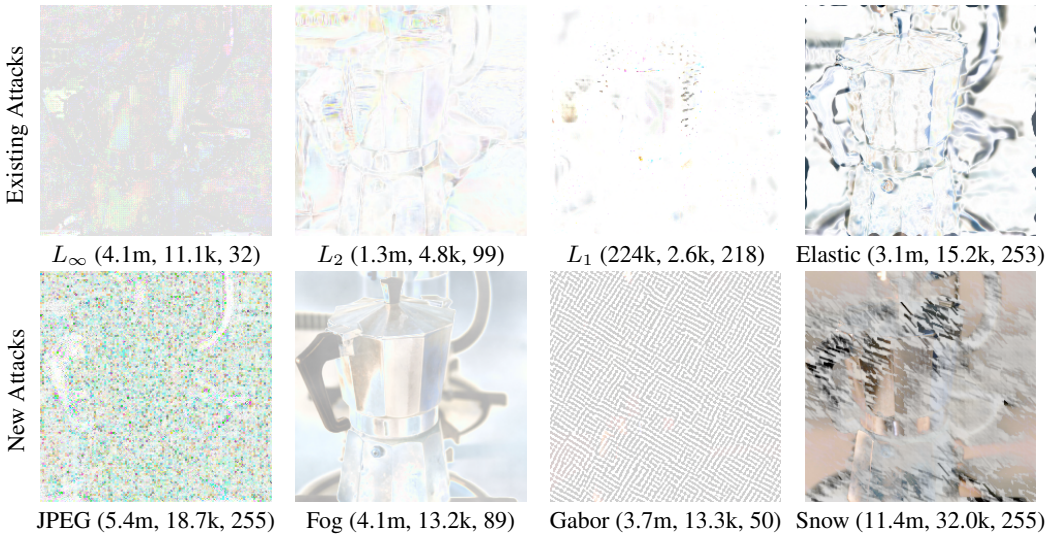


Figure 9: Differences of the attacked images and original image for different attacks (label "espresso maker"). The L_1, L_2, and L_∞ norms of the difference are shown in parentheses. As shown, our novel attacks display qualitatively different behavior and do not fall under the L_p threat model. These differences are not scaled and are normalized so that no difference corresponds to white.

Algorithm 1 Pseudocode for the Frank-Wolfe algorithm for the L_1 attack.

Input: function f, initial input $x \in [0,1]^d$, L_1 radius ρ, number of steps T.
Output: approximate maximizer $\bar{x}$ of f over the truncated L_1 ball $B_1(\rho; x) \cap [0,1]^d$ centered at x.

$x^{(0)} \leftarrow \mathrm{RandomInit}(x)$ ▷ Random initialization
for $t = 1, \ldots, T$ **do**
  $g \leftarrow \nabla f(x^{(t-1)})$ ▷ Obtain gradient
  for $k = 1, \ldots, d$ **do**
    $s_k \leftarrow$ index of the coordinate of g by with k^{th} largest norm
  end for
  $S_k \leftarrow \{s_1, \ldots, s_k\}$.

  for $i = 1, \ldots, d$ **do** ▷ Compute move to boundary of $[0,1]$ for each coordinate.
    if $g_i > 0$ **then**
      $b_i \leftarrow 1 - x_i$
    else
      $b_i \leftarrow -x_i$
    end if
  end for
  $M_k \leftarrow \sum_{i \in S_k} |b_i|$ ▷ Compute L_1-perturbation of moving k largest coordinates.
  $k^* \leftarrow \max\{k \mid M_k \leq \rho\}$ ▷ Choose largest k satisfying L_1 constraint.
  for $i = 1, \ldots, d$ **do** ▷ Compute $\hat{x}$ maximizing $g^\top x$ over the L_1 ball.
    if $i \in S_{k^*}$ **then**
      $\hat{x}_i \leftarrow x_i + b_i$
    else if $i = s_{k^*+1}$ **then**
      $\hat{x}_i \leftarrow x_i + (\rho - M_{k^*}) \operatorname{sign}(g_i)$
    else
      $\hat{x}_i \leftarrow x_i$
    end if
  end for
  $x^{(t)} \leftarrow (1 - \frac{1}{t})x^{(t-1)} + \frac{1}{t}\hat{x}$ ▷ Average $\hat{x}$ with previous iterates
end for
$\bar{x} \leftarrow x^{(T)}$

Table 2: ATA values for L_1-JPEG and L_2-JPEG on ImageNet-100.

Attack	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ATA_1	ATA_2	ATA_3	ATA_4	ATA_5	ATA_6
L_2-JPEG	8	16	32	64	128	256	84.8	82.5	78.9	72.3	47.5	3.4
L_1-JPEG	256	1024	4096	16384	65536	131072	84.8	81.8	76.2	67.1	46.4	41.8

we find that they have extremely similar results, so we omit L_1-JPEG in the full analysis for brevity and visibility. Calibration values for these attacks are shown in Table 2.

C.2 Full evaluation results and analysis for ImageNet-100

We show the full results of all adversarial attacks against all adversarial defenses for ImageNet-100 in Figure 11. As described, the L_p attacks and defenses give highly correlated information on held-out defenses and attacks respectively. Thus, we recommend evaluating on a wide range of distortion types. Full UAR scores are also provided for ImageNet-100 in Figure 12.

We further show selected results in Figure 13. As shown, a wide range of ε is required to see the full behavior.

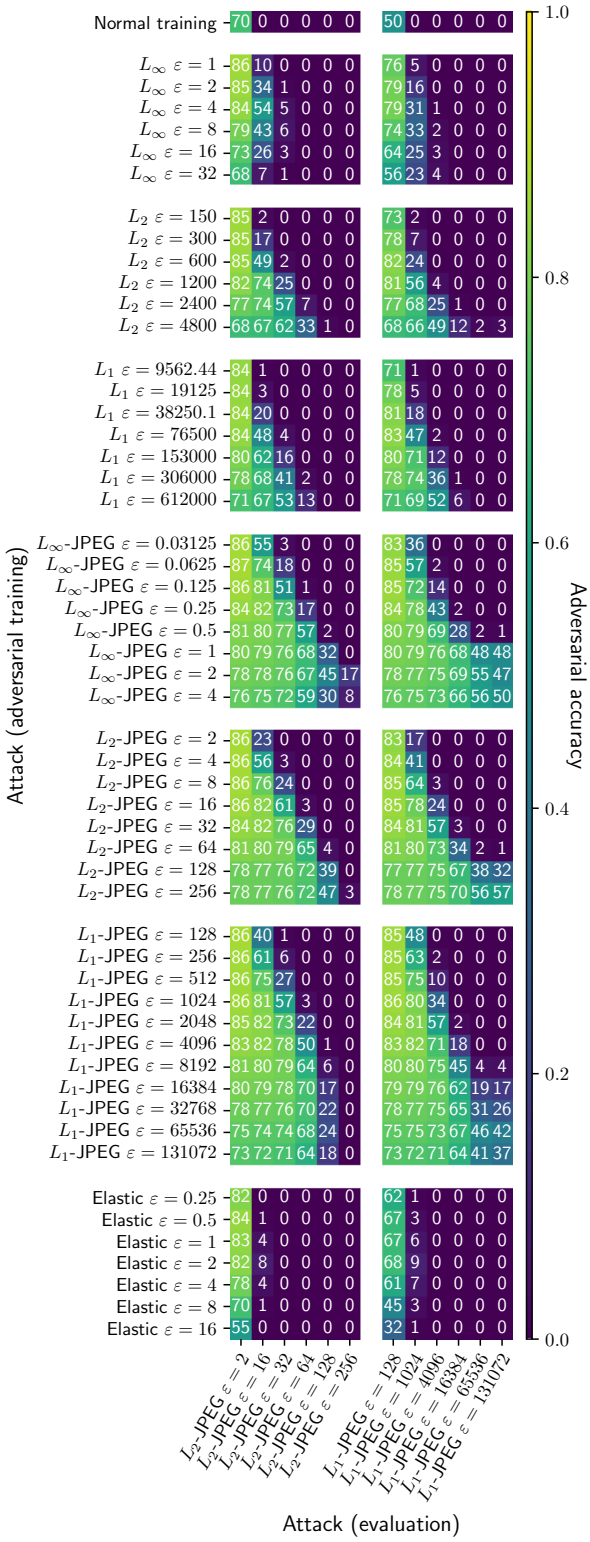


Figure 10: A comparison of L_1-JPEG and L_2-JPEG attacks.

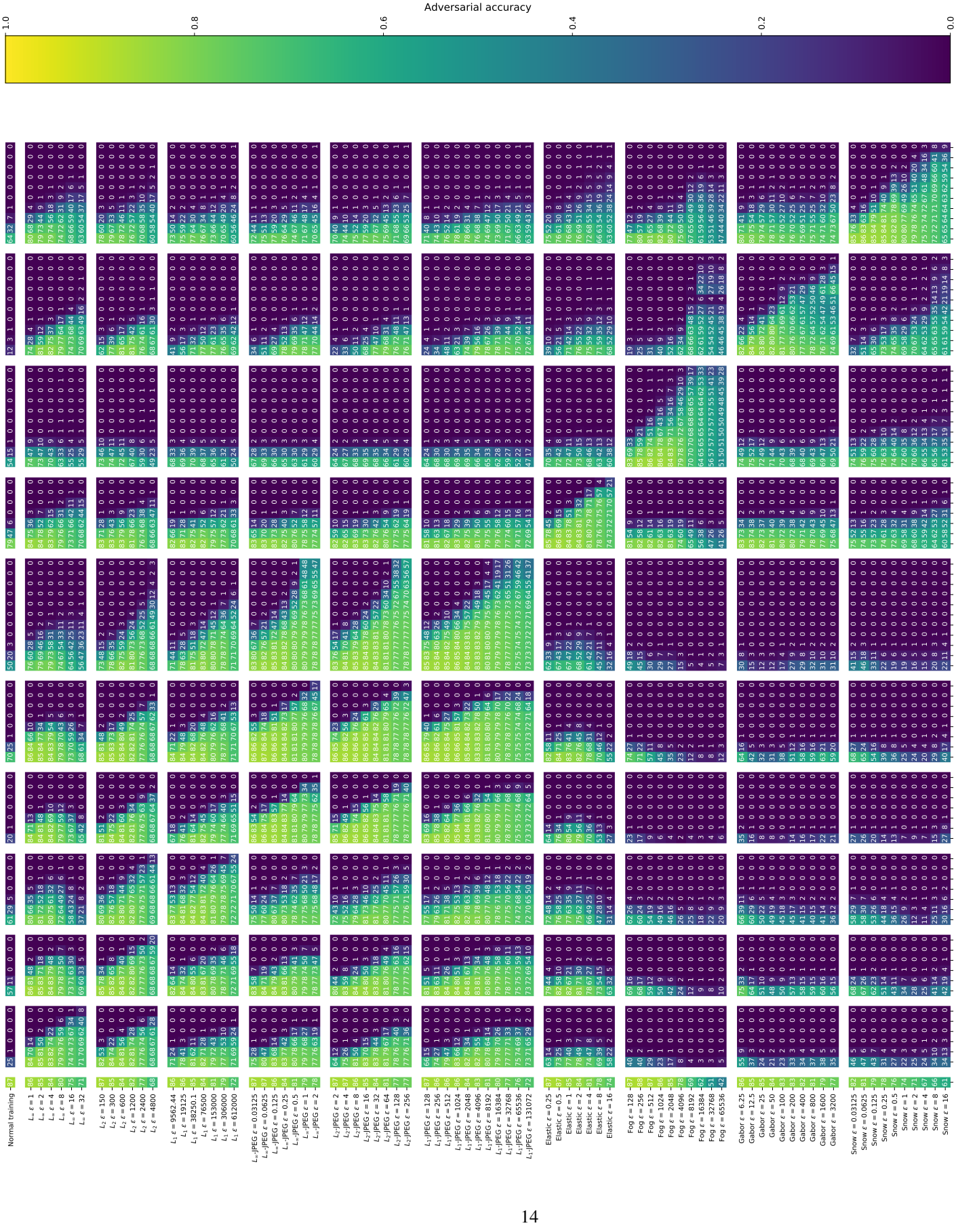


Figure 11: Accuracy of adversarial attack (column) against adversarially trained model (row) on ImageNet-100.

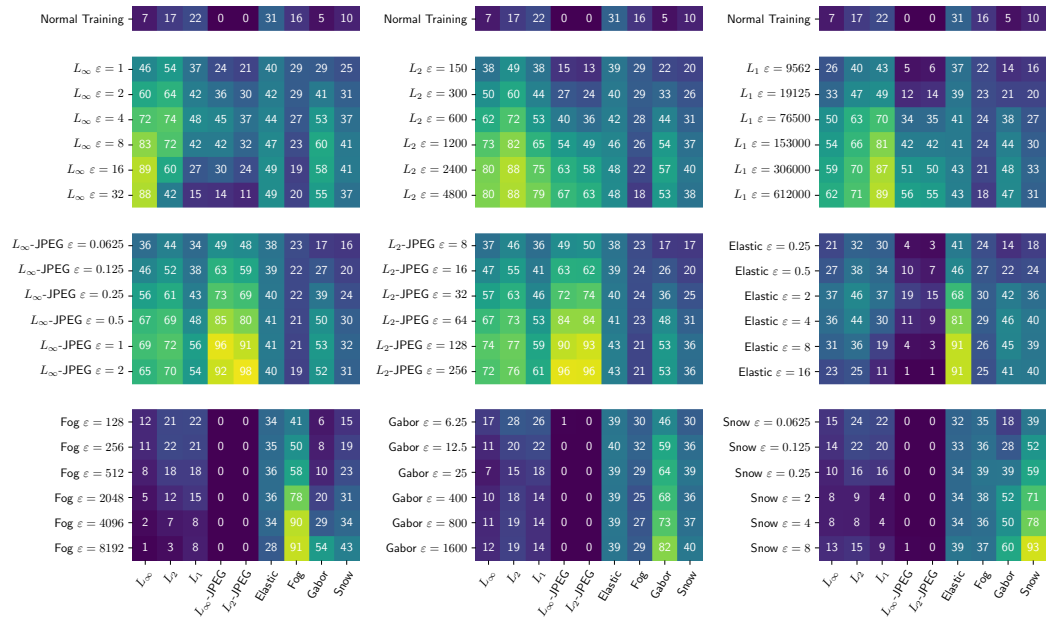


Figure 12: UAR scores (multiplied by 100) for adv. trained defenses (rows) against distortion types (columns) for ImageNet-100.

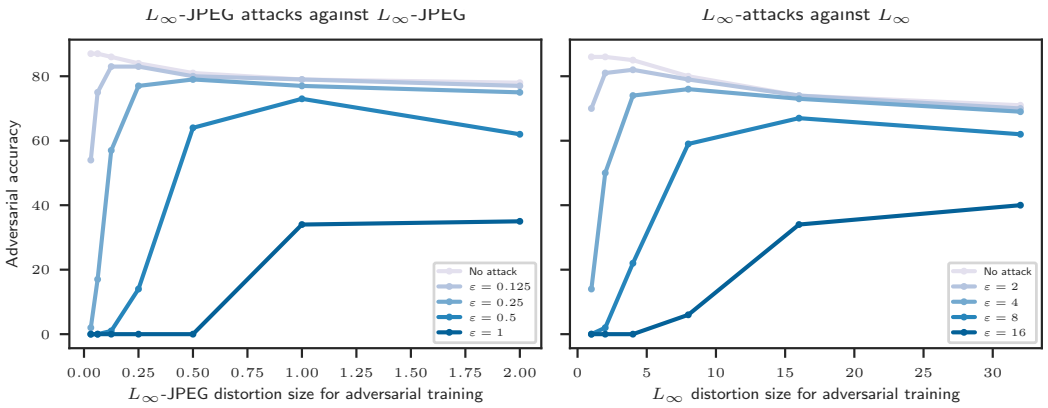


Figure 13: Adversarial accuracies of attacks on adversarially trained models for different distortion sizes on ImageNet-100. For a given attack ε, the best ε' to train against satisfies $\varepsilon' > \varepsilon$ because the random scaling of ε' during adversarial training ensures that a typical distortion during adversarial training has size smaller than ε'.

C.3 Full evaluation results and analysis for CIFAR-10

C.3.1 Full results for CIFAR10

We show the results of adversarial attacks and defenses for CIFAR-10 in Figure 14. We experienced difficulty training the L_2 and L_1 attacks at distortion sizes greater than those shown and have omitted those runs, which we believe may be related to the small size of CIFAR-10 images.

C.3.2 ATA and UAR for CIFAR-10

The ε calibration procedure for CIFAR-10 was similar to that used for ImageNet-100. We started with the perceptually small ε_{min} values in Table 3 and increased ε geometrically with ratio 2 until adversarial accuracy of an adversarially trained model dropped below 40. Note that this threshold

Table 3: Calibrated distortion sizes and ATA values for ResNet-56 on CIFAR-10

Attack	ε_1	ε_2	ε_3	ε_4	ε_5	ε_6	ATA_1	ATA_2	ATA_3	ATA_4	ATA_5	ATA_6
L_∞	1	2	4	8	16	32	91.0	87.8	81.6	71.3	46.5	23.1
L_2	40	80	160	320	640	2560	90.1	86.4	79.6	67.3	49.9	17.3
L_1	195	390	780	1560	6240	24960	92.2	90.0	83.2	73.8	47.4	35.3
L_∞-JPEG	0.03125	0.0625	0.125	0.25	0.5	1	89.7	87.0	83.1	78.6	69.7	35.4
L_1-JPEG	2	8	64	256	512	1024	91.4	88.1	80.2	68.9	56.3	37.7
Elastic	0.125	0.25	0.5	1	2	8	87.4	81.3	72.1	58.2	45.4	27.8

is higher for CIFAR-10 because there are fewer classes. The resulting ATA and UAR values for CIFAR10 are shown in Table 3 and Figure 15. We omitted calibration for the L_2-JPEG attack because we chose too small a range of ε for our initial training experiments, and we plan to address this issue in the future.

D Robustness of our results

D.1 Replication

We replicated our results for the first three rows of Figure 11 with different random seeds to see the variation in our results. As shown in Figure 16, deviations in results are minor.

D.2 Convergence

We replicated the results in Figure 11 with 50 instead of 200 steps to see how the results changed based on the number of steps in the attack. As shown in Figure 17, the deviations are minor.

E Further Results for Joint Training

E.1 Full experimental results

We show the evaluation accuracies of jointly trained models in Figure 18.

We show all the attacks against the jointly adversarially trained defenses in Figure 19.

E.2 Dependence on random seed

In Table 4, we study the dependence of joint adversarial training to random seed. We find that at large distortion sizes, joint training for certain pairs of distortions does not produce consistent results over different random initializations.

Table 4: Train and val accuracies for joint adversarial training at large distortion are dependent on seed. For train and val, ε' is chosen uniformly at random between 0 and ε, and we used 10 steps for L_∞ and L_1 and 30 steps for elastic. Single adversarial training baselines are also shown.

Training parameters (ResNet-50)	L_∞ train	other train	L_∞ val	other val
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$, Seed 1	90	89	35	74
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$, Seed 2	89	90	47	44
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$, Seed 3	90	89	29	63
L_∞ $\varepsilon = 16$, L_1 $\varepsilon = 612000$, Seed 1	86	87	22	16
L_∞ $\varepsilon = 16$, L_1 $\varepsilon = 612000$, Seed 2	88	87	16	24
L_∞ $\varepsilon = 8$	81	–	74	–
L_∞ $\varepsilon = 16$	68	–	63	–
Elastic $\varepsilon = 4$	–	88	–	76
L_1 $\varepsilon = 612000$	–	75	–	59

Table 5: Training and validation numbers for ResNet-101 and ResNet-50 for joint training against L_∞, $\varepsilon = 8$ and elastic, $\varepsilon = 4$.

Training parameters	L_∞ train	other train	L_∞ val	other val
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$, ResNet-50 Seed 1	90	89	35	74
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$, ResNet-50 Seed 2	89	90	47	44
L_∞ $\varepsilon = 8$, Elastic $\varepsilon = 4$ ResNet-101	90	91	49	46

E.3 Overfitting and model capacity

As a first test to understand the relationship between model capacity and overfitting, we trained ResNet-101 models using the same procedure as in Section 5. Briefly, overfitting still occurs, but ResNet-101 achieves a few percentage points higher than ResNet-50.

We show the training curves in Figure 20 and the training and validation numbers in Table 5.

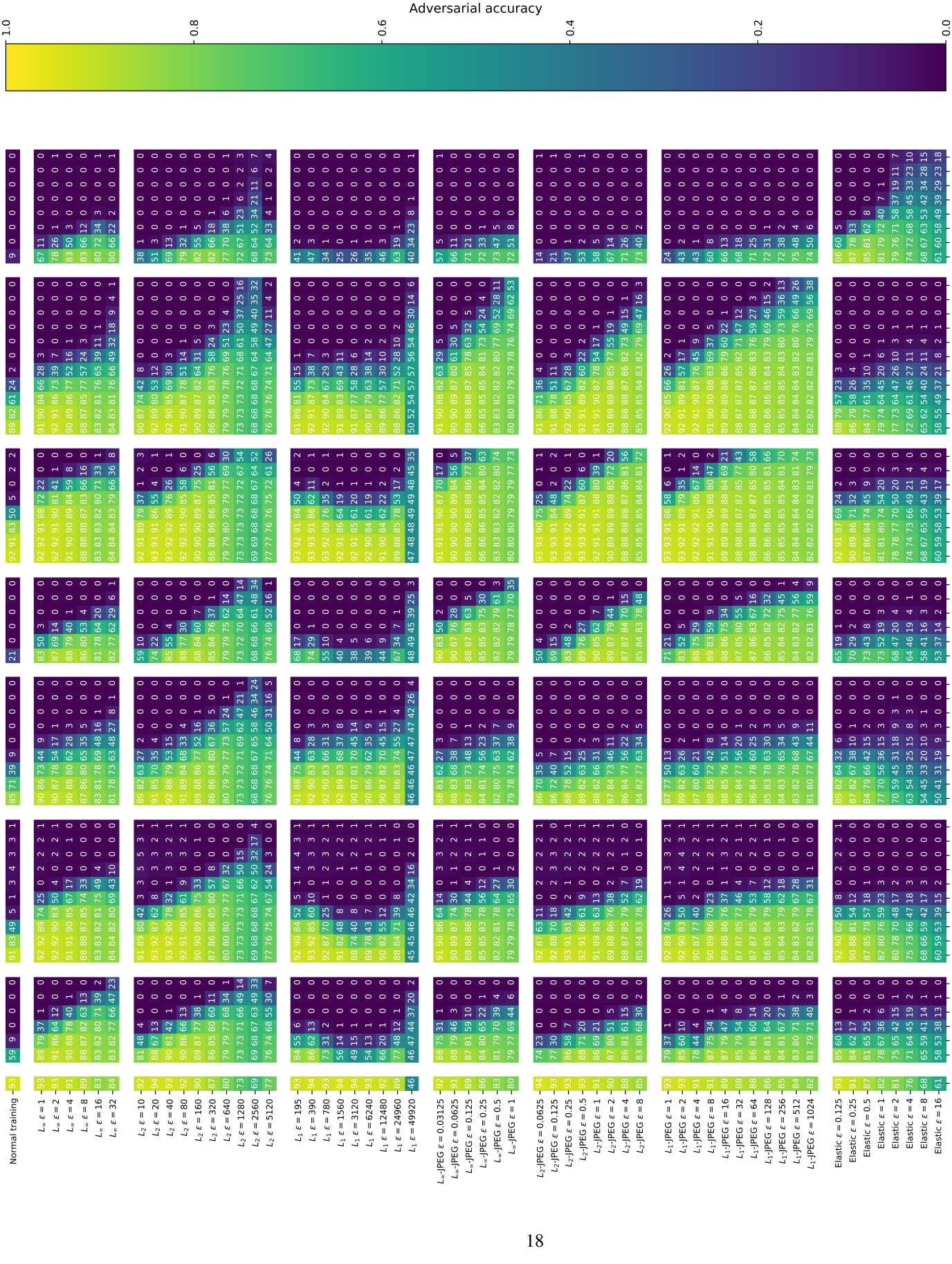


Figure 14: Accuracy of adversarial attack (column) against adversarially trained model (row) on CIFAR-10.

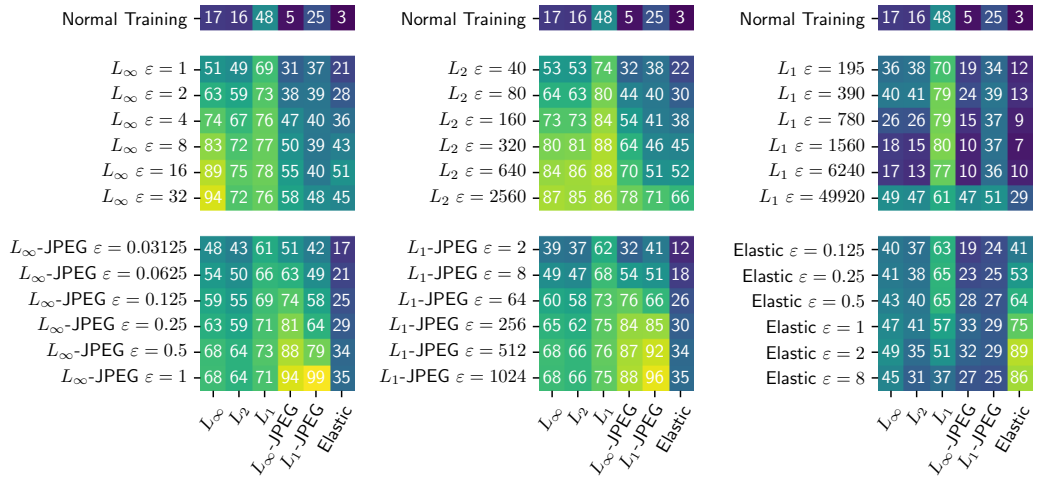


Figure 15: UAR scores on CIFAR-10. Displayed UAR scores are multiplied by 100 for clarity.

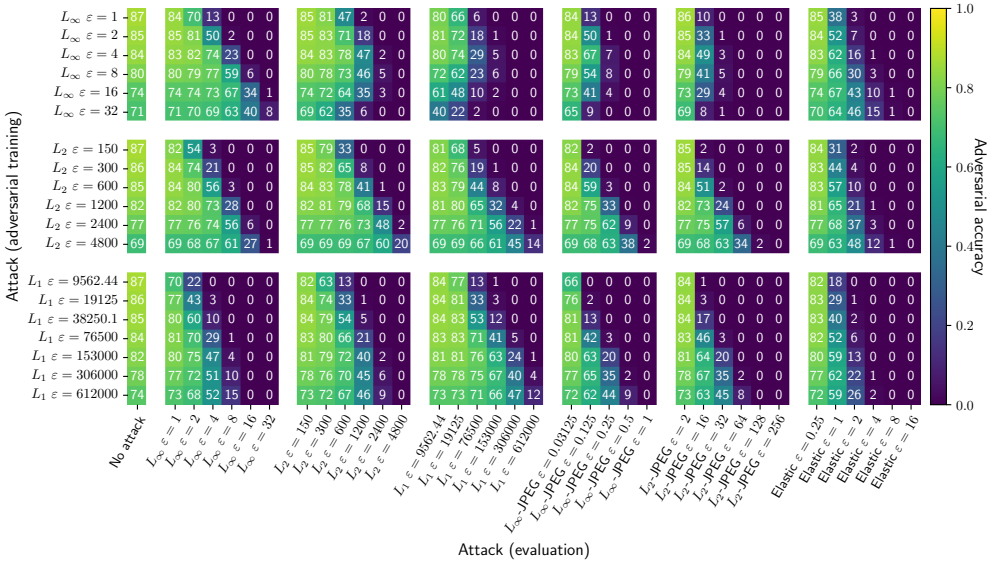


Figure 16: Replica of the first three block rows of Figure 11 with different random seeds. Deviations in results are minor.

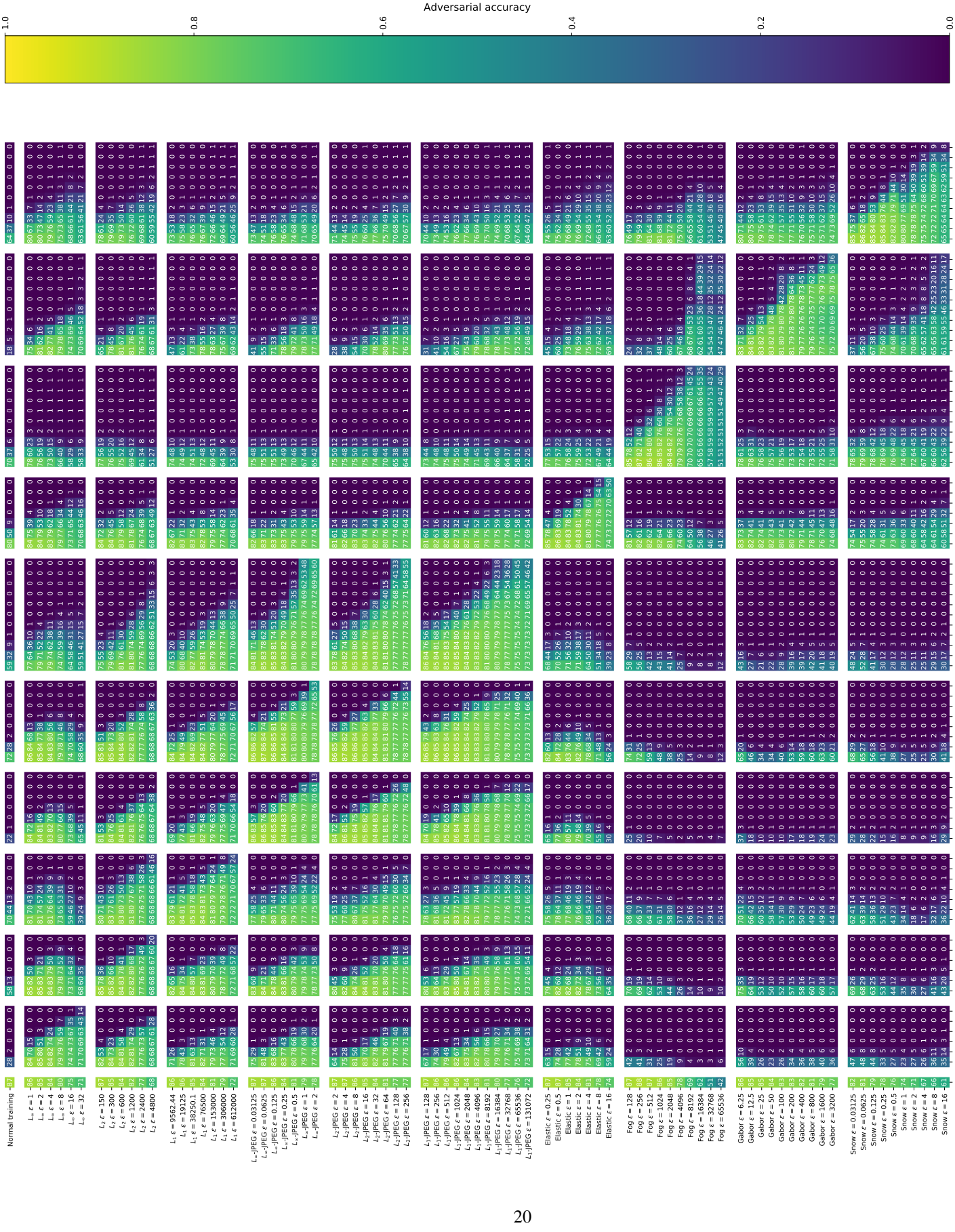


Figure 17: Replica of Figure 11 with 50 steps instead of 200 at evaluation time. Deviations in results are minor.

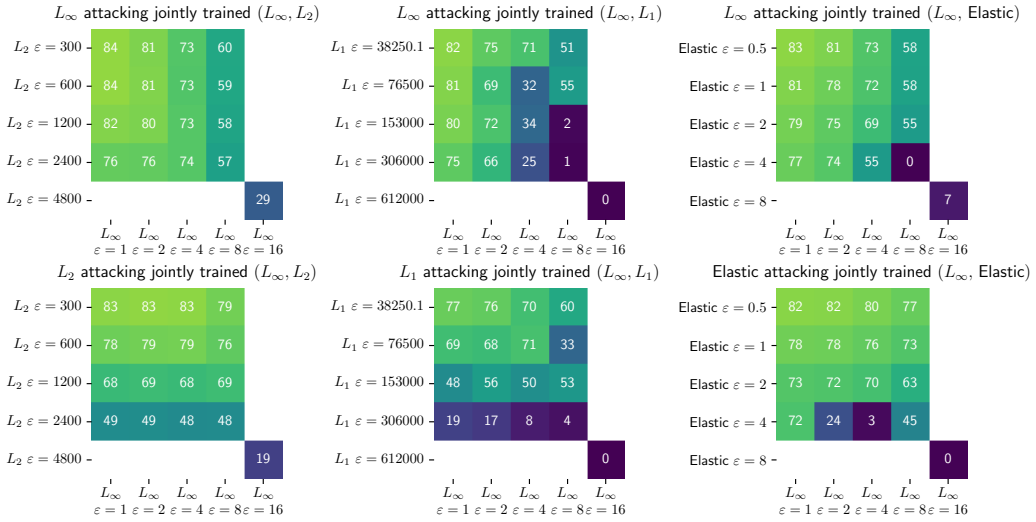


Figure 18: Evaluation accuracies of jointly trained models. Attack and training ε values are equal.

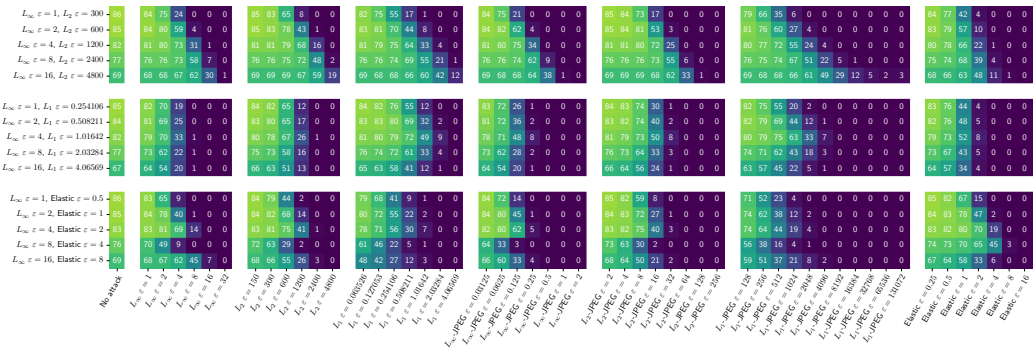


Figure 19: All attacks (columns) vs. jointly adversarially trained defense (rows).

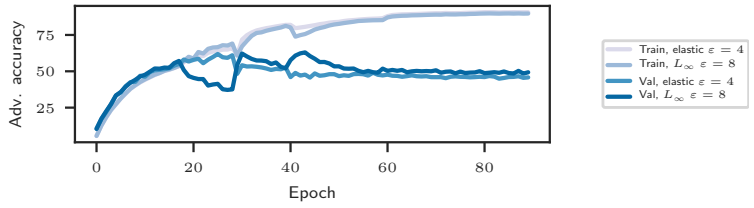


Figure 20: Train and validation curves for joint training against L_∞, $\varepsilon = 4$ and elastic, $\varepsilon = 8$ using ResNet-101. As shown, the validation accuracies decrease as training progresses, indicating overfitting.