MultiRobustBench: Benchmarking Robustness Against Multiple Attacks

Sihui Dai [1] **Saeed Mahloujifar** [1] **Chong Xiang** [1] **Vikash Sehwag** [1] **Pin-Yu Chen** [2] **Prateek Mittal** [1]

Abstract

The bulk of existing research in defending against adversarial examples focuses on defending against a single (typically bounded ℓ_p-norm) attack, but for a practical setting, machine learning (ML) models should be robust to a wide variety of attacks. In this paper, we present the first unified framework for considering multiple attacks against ML models. Our framework is able to model different levels of learner's knowledge about the test-time adversary, allowing us to model robustness against unforeseen attacks and robustness against unions of attacks. Using our framework, we present the first leaderboard, MultiRobustBench (`https://multirobustbench.github.io`), for benchmarking multiattack evaluation which captures performance across attack types and attack strengths. We evaluate the performance of 16 defended models for robustness against a set of 9 different attack types, including ℓ_p-based threat models, spatial transformations, and color changes, at 20 different attack strengths (180 attacks total). Additionally, we analyze the state of current defenses against multiple attacks. Our analysis shows that while existing defenses have made progress in terms of average robustness across the set of attacks used, robustness against the worst-case attack is still a big open problem as all existing models perform worse than random guessing.

1. Introduction

For safety-critical applications, it is important that machine learning (ML) models are robust against test-time adversaries. These test-time adversaries can potentially use multiple (and unforeseen) attack types, motivating the need to study multiattack robustness. Several works (Maini et al., 2020; Tramèr & Boneh, 2019; Croce & Hein, 2020a; Dai et al., 2022; Jin & Rinard, 2020; Laidlaw et al., 2021; Hsiung et al., 2022a) design defenses for multiattack robustness, but these works lack a unified evaluation framework: these works utilize different small sets of attacks and attack strengths. The lack of a standardized benchmark for evaluating multiattack robustness is an obstacle to understanding and improving upon the current progress made by the community towards multiattack robustness.

To improve our understanding of current progress in multiattack robustness, we introduce MultiRobustBench (available at `multirobustbench.github.io`), which provides a leaderboard for multiattack robustness based on two new metrics that we introduce: *competitiveness ratio* (CR) and *stability constant* (SC). CR measures how close the robust accuracy of a defense on each attack type is to the robust accuracy of the best performing model for each specific attack type. SC measures robustness degradation across attacks of different strengths. Our benchmark evaluates 16 defended models based on a set of 9 different attacks across 20 levels of attack strengths (180 attacks total, 2880 evaluations overall), making it the largest multiattack evaluation to date. Our benchmark allows us to draw important insights on the state of research in multiattack robustness; specifically, we find that while existing research has made progress on average robustness over this set of attacks, all existing defenses perform worse than random guessing in worst-case multiattack robustness.

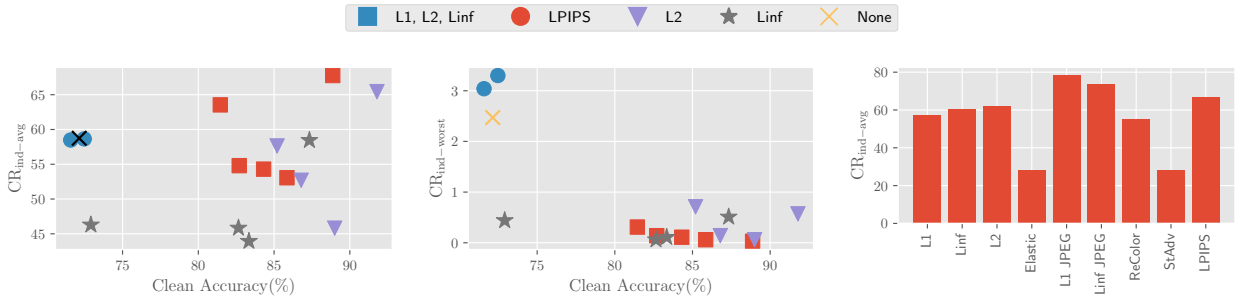
Our contributions are as follows:

We introduce an adversarial game framework for multiattack robustness. This framework unifies previously studied settings such as robustness against unions of known attacks and robustness against unforeseen attacks by introducing *knowledge sets* which capture mismatch in threat models used during training and test-time. Using this framework, we define a taxonomy of settings in multiattack robustness.

We introduce metrics (competitiveness ratio and stability constant) for measuring multiattack performance. Competitiveness ratio (CR) can be interpreted as an aggregated percentage representing how close the accuracy of the defense is to the accuracy of the best performing models,

[1]Electrical and Computer Engineering, Princeton University [2]IBM Research. Correspondence to: Sihui Dai <sihuid@princeton.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.



(a) Clean accuracy vs $\text{CR}_{\text{ind-avg}}$ (b) Clean accuracy vs $\text{CR}_{\text{ind-worst}}$ (c) Average single-$\text{CR}_{\text{ind-avg}}$

Figure 1: **State of current defenses.** Figures (a) and (b): Clean accuracy and CR of existing techniques on our leaderboard in terms. Each marker represents a single defense and the marker shape/color represents the types of attacks used by the defense during training (K_{learner}). Figure (c): $\text{CR}_{\text{ind-avg}}$ taken across each attack type averaged across all 16 defenses.

while stability constant (SC) measures how much robustness changes when switching to a different attack strengths. We introduce 2 different variants of CR, one for measuring average multiattack robustness ($\text{CR}_{\text{ind-avg}}$) and one for measuring worst case multiattack robustness $\text{CR}_{\text{ind-worst}}$.

Using our proposed metrics, we provide a leaderboard containing evaluations of existing defenses targeting muliattack robustness. Our leaderboard evaluates existing models on a wide variety of attacks including bounded ℓ_p norm attacks, color changes (Laidlaw et al., 2021), spatial transformations (Xiao et al., 2018), elastic attacks, JPEG attacks (Kang et al., 2019), and bounded LPIPS attacks (Laidlaw et al., 2021). Our leaderboard also provides features such as performance visualizations, which can be a useful diagnostic tool for understanding weaknesses of individual defenses.

We analyze the state of current defenses for multiattack robustness. We find that while current models have decent performance in terms of average case robustness across multiple (imperceptible) attacks, there is significant room for improvement in terms of worst-case performance. Additionally, using our metrics, we analyze how factors such as architecture size, use of additional training data, and number of training epochs influence multiattack robustness.

Overall, benchmarks have the potential to revolutionize ML by enabling comparable research and highlighting an open research problem for our community. We hope that our benchmark inspires research in multiattack robustness and accelerates the development of stronger defenses.

2. Prior Work

Adversarial robustness. Prior works have demonstrated a vulnerability of existing ML models: imperceptible perturbations during test-time can cause models to misclassify (Szegedy et al., 2014). These imperceptible perturbations can be of many different types including norm-bounded perturbations, small spatial transformations (Xiao et al., 2018), color changes (Laidlaw & Feizi, 2019), and their compositions (Hsiung et al., 2022b). Although a variety of attack types exist, the majority of current research in adversarial robustness focuses on defending models against perturbations that are bounded in ℓ_2 or ℓ_∞ norm. Adversarial training (Madry et al., 2018; Zhang et al., 2019; Gowal et al., 2020), a popular defense framework in which the model is trained with adversarial examples, has been mainly studied for ℓ_p robustness. For example, prior works have studied the impact of architecture size (Wu et al., 2021; Huang et al., 2021), early stopping (Rice et al., 2020), and additional training data (Carmon et al., 2019; Rebuffi et al., 2021; Gowal et al., 2021; Sehwag et al., 2021) on robustness on the ℓ_∞ and ℓ_2 attacks used during adversarial training. However, it is unclear how applicable these findings and defenses are in practice, where the adversary can potentially use multiple attacks which might not be known in advance to the defender. In our work, we find that general trends observed for robustness against single (known) attacks do not necessarily hold for multiattack robustness.

Multiattack robustness. Several prior works have studied robustness against multiple attacks. One line of works focuses on improving robustness against the union of known attacks (typically the union of ℓ_p-balls) (Maini et al., 2020; Tramèr & Boneh, 2019; Croce & Hein, 2020a; Madaan et al., 2020). Another line of works looks at defending against attacks that are not used during training (Laidlaw et al., 2021; Dai et al., 2022; Jin & Rinard, 2020). We provide a framework that unifies both of these research directions and provides metrics and a leaderboard for benchmarking these defenses.

Benchmarking adversarial robustness. RobustBench (Croce et al., 2020) is a standardized benchmark for ad-

versarial robustness which provides leaderboards for ℓ_∞ and ℓ_2 robustness measured via AutoAttack (Croce & Hein, 2020b). CARBEN (Hsiung et al., 2022b) is a benchmark for measuring robustness against compositions of ℓ_∞ attacks with global spatial transformations (i.e. rotation) and global color shifts (i.e. hue shift). These global transformations are weaker than existing attacks like StAdv (Xiao et al., 2018) and ReColor (Laidlaw & Feizi, 2019) which optimize over pixels, making it unclear how well evaluations from CARBEN reflect true multiattack robustness.

As a research community, we lack a standardized benchmark that accurately reflects robustness against multiple attacks. Current works in multiattack robustness deploy different methods for evaluating the performance of their defense, such as reporting accuracies for different sets of attacks or using custom metrics (such as mUAR (Kang et al., 2019)) for measuring performance. To standardize evaluation for multiattack robustness, we introduce a leaderboard which ranks existing defenses for multiattack robustness based on metrics motivated by our proposed framework for multiattack robustness. Our leaderboard also provides performance visualizations (Appendix F) for each defense so that researchers can understand the weaknesses of existing defenses in detail, which can potentially lead to improvements in the direction of multiattack robustness.

3. Robustness against multiple attacks

We begin by first providing a framework for modeling problems in multiattack robustness. We then discuss goals in multiattack robustness and metrics for measuring it.

Notations We use $\mathcal{D} = X \times Y$ to denote the data distribution where X is the support of the images and Y is the support of the labels. We use D_{train} to denote the training set with data sampled from $\mathcal{D}$ in an i.i.d. manner. We will refer to the defense as a learning algorithm which we will denote with $\mathcal{A}$. We use $\mathcal{H}$ to denote the hypothesis class used by $\mathcal{A}$ (ie. $\mathcal{A}$ outputs a function $h \in \mathcal{H}$).

3.1. A unified adversarial game framework for modeling robustness against multiple attacks

While prior works have studied problems relating to robustness against multiple attacks (Tramèr & Boneh, 2019; Maini et al., 2020; Croce & Hein, 2020a; Laidlaw et al., 2021; Dai et al., 2022; Jin & Rinard, 2020; Hsiung et al., 2022a), these works have studied specific instances of multiattack robustness (i.e. robustness against unions of known threat models, unforeseen attack robustness), but have not provided a unified framework for modelling problems in multiattack robustness. In this section, we propose a unified framework for multiattack robustness by providing an adversarial game formulation.

We begin by introducing a perturbation function which maps inputs to adversarial examples.

Definition 3.1 (Perturbation Function)**.** Let $C : X \to 2^X$ define the constraint of the adversary and $\ell : Y \times Y \to \mathbb{R}$ be a loss function. A perturbation function $P_C : X \times Y \times \mathcal{H} \to X$ maps input and hypothesis to adversarially perturbed versions of the input:

$$P_C(x, y, h) = \arg\max_{x' \in C(x)} \ell(h(x), y)$$

To capture multiattack settings such as robustness against unforeseen attacks where the learner does not know what type of attacks are present during test time, we introduce a knowledge set.

Definition 3.2 (Knowledge Set)**.** A knowledge set K_{learner} is a set of perturbation functions. We say that the defender is restricted to knowledge set K_{learner} if the learning algorithm optimizes model selection by using information about perturbation functions only within K_{learner}.

The learner and attacker knowledge sets allow us to model robustness against multiple perturbations as an adversarial game:

Definition 3.3 (Adversarial Game for Multiple Attacks)**.**

1. Environment specifies a robustness threshold γ and specifies a (possibly infinite) set K of perturbation functions that can occur during test-time. The environment also specifies the learner's knowledge set K_{learner} where $|K_{\text{learner}}| \leq |K|$.
2. The learner then chooses learning algorithm $\mathcal{A}$ and obtains model $h = \mathcal{A}(D_{\text{train}}, K_{\text{learner}})$. Here, $\mathcal{A}(D_{\text{train}}, K_{\text{learner}})$ denotes that the learning algorithm is restricted to using information about perturbation functions within K_{learner}.
3. If $\frac{\text{err}_{\text{multi}}(h;K)}{\min_{h^* \in \mathcal{H}} \text{err}_{\text{multi}}(h^*;K)} \leq \gamma$, then the learner wins and $\mathcal{A}$ produces a model that is close to optimal against K. Otherwise the attacker wins.

The definition of $\text{err}_{\text{multi}}$ and relationship between K and K_{learner} can lead to different forms of robustness against multiple attacks.

Relationship between K and K_{learner}. The relationship between K and K_{learner} leads to different settings for robustness against multiple attacks. The setting where $K = K_{\text{learner}}$ models the commonly studied setting where the learner knows the attacks used during evaluation in advance and can optimize their model directly with respect to those attacks. For example, works studying robustness against unions of ℓ_p attacks (Tramèr & Boneh, 2019; Maini et al., 2020; Croce & Hein, 2020a) fall under this category. We call the setting where $K = K_{\text{learner}}$ the *full knowledge* setting. We note that when $|K| = 1$, the adversarial game for

multiple attacks reduces to the adversarial game for a single attack.

When $K \neq K_{\text{learner}}$, there is a mismatch between the attacks that the learner is aware about and can use for the learning algorithm. We call this the *knowledge mismatch* setting. The types of mismatches can be divided into several cases: 1) $K \cap K_{\text{learner}} = \emptyset$, 2) $K \cap K_{\text{learner}} \neq \emptyset$.

The first case represents settings where the learner has no knowledge of the true space of attacks. An example of this is if the adversary is constrained to using patch attacks, while the learner is under the impression that there will only be imperceptible attacks at test-time so K_{learner} consists of a selection of imperceptible attacks (ie. bounded ℓ_p attacks). We call this the setting of *no knowledge*.

The second case represents settings where the learner knows only a subset of attacks that will be used during test-time. We call this setting the *partial knowledge* setting and contains the problem of unforeseen attacks. An example of this is when the test-time adversary is restricted to attacks that do not change a human's classification of the image, but the learner does not know how to model the full space of these attacks and is aware of only a subset of those attacks (ie. bounded ℓ_p attacks). For the task of image classification, the setting of learner knowledge models a more realistic learning setting compared to the full knowledge since we would like our model to be robust against attacks developed in the future. The partial knowledge setting is also more realistic in comparison to the no knowledge setting for image classification since existing known attacks are also valid attacks that can be used the adversary during test-time.

In Appendix A, we categorize existing defenses against multiple attacks into full, partial, and no knowledge settings.

Definition of $\text{err}_{\text{multi}}$. Choosing the definition of err for multiple attacks also leads to different problems in multiattack robustness. For example, if the learner knows the distribution $\mathcal{P}(K)$ of frequency at which attacker chooses each attack in K, then this can be modeled with $\text{err}_{\text{multi-exp}}(h; K) := \mathbb{E}_{P\sim\mathcal{P}(K)}\text{err}(h; P)$. The learner can also consider using the worst case error across all $P \in K$ as a measure of multiattack performance: $\text{err}_{\text{multi-max}}(h; K) := \max_{P\in K} \text{err}(h; P)$.

Another possibility is to let γ be a vector of length equal to the number of perturbation functions in K and letting $\text{err}_{\text{multi}}$ output a vector of errors with respect to each individual perturbation $P \in K$ (ie. $\text{err}_{\text{multi-ind}} := [\text{err}(h; P)]_{P\in K}$). In this case, the learner only wins the game if the losses on each individual attack lies within the corresponding robustness threshold in the vector γ. $\text{err}_{\text{multi-ind}}$ allows us to model the problem of achieving robustness against the union of attacks in K while also allowing us to specify how much tradeoff in performance across attacks we are willing to tolerate.

3.2. Metrics for evaluating multiattack robustness

Using the adversarial game formulation in Definition 3.3, we now design metrics which aggregate accuracy across each individual attack into a single number. In this section, we discuss two potential criteria that we would like to achieve when designing a good defense: 1) competitive performance and 2) stability across attack difficulty and introduce the metrics we use for measuring each criterion.

Competitive performance across attacks. In the multiattack adversarial game formulation in Definition 3.3, we saw that the objective of the learner is to choose a learning algorithm which allows the learner to obtain a model h whose performance is competitive with the best model in the hypothesis set with respect to the choice of $\text{err}_{\text{multi}}$. We introduce a family of metrics which we call competitiveness ratio (CR), which measures how close h is to the best model in the hypothesis set.

Definition 3.4 (Competitiveness Ratio (CR))**.** Let $\text{acc}^*_{\text{multi}}(K) := 1 - \min_{h^*\in\mathcal{H}} \text{err}_{\text{multi}}(h^*; K)$ and $\text{acc}_{\text{multi}}(h, K) := 1 - \text{err}_{\text{multi}}(h; K)$. Then, the competitiveness ratio (CR) of a defended model h is given by:

$$\text{CR}(h; K) = 100 \times \frac{\text{acc}_{\text{multi}}(h, K)}{\text{acc}^*_{\text{multi}}(K)} \tag{1}$$

In practice, we approximate acc^* through adversarial training and will discuss this in more depth in Section 4.1. We note that CR can be used in all knowledge settings since metrics are taken with respect to K which can differ from K_{learner}. Using different definitions of $\text{err}_{\text{multi}}$ leads to different variants of CR.

For example, if we use $\text{err}_{\text{multi-ind}}$ as the multi-attack error function, then CR compares each attack within K to the best accuracy on that specific attack. We can then aggregate all of these scores by either taking the expectation or worst case, leading to the following variants of CR:

Definition 3.5. ($\text{CR}_{\text{ind-avg}}$ and $\text{CR}_{\text{ind-worst}}$) For a single $P \in K$, let $\text{acc}^*(P) := 1 - \min_{h\in\mathcal{H}} \text{err}(h; P)$ and $\text{acc}(h, P) := 1 - \text{err}(h; P)$. Then,

$$\text{CR}_{\text{ind-avg}}(h; K) := 100 \times \mathbb{E}_{P\sim\mathcal{P}(K)} \left[\frac{\text{acc}(h, P)}{\text{acc}^*(P)}\right] \tag{2}$$

$$\text{CR}_{\text{ind-worst}}(h; K) := 100 \times \min_{P\in K} \frac{\text{acc}(h, P)}{\text{acc}^*(P)} \tag{3}$$

We discuss using other choices for $\text{err}_{\text{multi}}$ in Appendix E.

For choices of $\text{err}_{\text{multi}}$, high CR indicates that the model h is closer to optimal with regards to our chosen definition of $\text{err}_{\text{multi}}$. When K contains only attacks of the same type (ie. ℓ_2 perturbations) at different strengths (ie. radii of ℓ_2 ball), we call this metric *single-CR*. When K contains

other attack types. Comparing single-CR values for a set of attacks allow us to understand whether there are some specific attack types that the model performs poorly on.

Stability across attack strength. As discussed in Section 3.1, there can exist a knowledge mismatch between K and K_{learner}. For example, K_{learner} can ℓ_2 perturbations with radius up to 0.50 while K contains ℓ_2 perturbations with radius up to 0.51. In this case, another goal of the learner is to have a graceful degradation of robustness in the vicinity of attacks in K_{learner}: since 0.51 is close to 0.50, we should not see a drastic difference in robustness from K_{learner} to K.

We now define an attack strength function, which measures difficulty of attacks. We will then use this definition to define *stability*, which is our criterion for measuring smooth degradation of robustness across attacks of similar difficulty.

Definition 3.6 (Attack strength function)**.** An *attack strength function* $s : K \to \mathbb{R}^+$ maps perturbation functions in attack set K to a number representing the difficulty of the attack. For example, if K contains a single attack type at different perturbation sizes ϵ, we can consider s to output the value of ϵ corresponding to the attack. As another example, with multiple attack types, we can consider an attack strength function $s(P) = \min_{h \in \mathcal{H}} \text{err}(h; P)$.

Using the attack strength function definition, we now define stability across perturbations.

Definition 3.7 (Stability across perturbations)**.** A model h is (L, α)-locally stable across perturbations with respect to attack strength function s if we have that for all $P_1 \in K_{\text{learner}}$ and $P_2 \in K$ such that $|s(P_1) - s(P_2)| \leq \alpha$, $|\text{acc}(h, P_1) - \text{acc}(h, P_2)| \leq L|s(P_1) - s(P_2)|$. Equivalently, for a given α and model h, we can compute the corresponding constant L, which we call the *stability constant (SC)* as follows:

$$L_\alpha(h) = \max_{\substack{P_1 \in K_{\text{learner}}, P_2 \in K \\ |s(P_1) - s(P_2)| \leq \alpha \\ P_1 \neq P_2}} \frac{|\text{acc}(h, P_1) - \text{acc}(h, P_2)|}{|s(P_1) - s(P_2)|} \quad (4)$$

In the above definitions of stability and SC, since α represents the difference in difficulty between attacks, we are interested in the regime of small α. Ideally, we would like SC to be small at small values of α, since that would suggest robust performance does not change much for attacks of similar difficulty.

4. Description of MultiRobustBench

Using CR and SC introduced in Section 3.2, we provide a leaderboard that ranks existing defenses against multiple adversarial perturbations in order to standardize evaluation of defenses against multiple attacks. Our leaderboard also provides visualizations of performance across individual attack types for researchers to analyze and understand strengths and weaknesses of their defenses. This leaderboard is available at https://multirobustbench.github.io/.

4.1. Evaluation Setup

Restrictions Similar to RobustBench (Croce et al., 2020), we focus on models that have a fully deterministic forward pass, nonzero gradients, and no optimization loop in the forward pass. Given that the bulk of attacks used in benchmarking are white-box attacks, our evaluations may be inaccurate for any model which does not satisfy these requirements.

Attack Space The space of attacks K which do not visually change the class of the original image is infinite, so it is important to define a subset of these attacks to use for evaluation. For benchmarking, we consider 9 different attack types at 20 different attack strengths (ϵ). We provide detailed descriptions of each attack and range of ϵ used per attack in Appendix C.

- **Bounded ℓ_p attacks** We consider ℓ_1, ℓ_2, and ℓ_∞ attacks. To measure robustness, we use apgd-t and fab-t from the AutoAttack package (Croce & Hein, 2020b). We restrict to using this subset of attacks to reduce evaluation time (since we evaluate a total of 60 ℓ_p attacks per model).
- **Color shift** For leaderboard rankings, we consider pixel-wise color shifts via ReColor attacks (Laidlaw & Feizi, 2019).
- **Spatial transformations** For leaderboard rankings, we consider small shifts in pixel positions using StAdv attacks (Xiao et al., 2018).
- **UAR attacks (Kang et al., 2019)** Kang et al. (2019) introduced a set of attacks for measuring unforeseen robustness including elastic, Linf JPEG, and L1 JPEG attacks. We incorporate these 3 attacks into our benchmark.
- **Bounded LPIPS attacks (Laidlaw et al., 2021)** Laidlaw et al. (2021) introduced 2 attacks (PPGD and LPA) based on LPIPS distance (Zhang et al., 2018), which is a more perceptually aligned distance metric than ℓ_p norms. For evaluation, we measure robustness against LPIPS attacks by taking the accuracy against the union of PPGD and LPA attacks.

Approximating optimal single attack accuracy. In Section 3.2, we defined CR in terms of optimal single attack accuracy $\text{acc}^*(P)$. In practice, we do not know these optimal values, so we approximate these by using the accuracies of ResNet-18 models that are trained using adversarial training directly on to attack of interest P. We choose to use ResNet-18 models for training efficiency and use the robust accuracy averaged over 3 runs for $\text{acc}^*(P)$. We note

Rank	Defense	Clean Acc	CR	SC	% Attacks Seen	Extra data	Architecture	Select
1	Learning to Generate Noise for Multi-Attack Robustness train with Linf eps=8/255, L2 eps=128/255, L1 eps=2000/255 with RST	88.87	67.76	1901.25	8.84	☑	WRN-28-10	☐
2	Fixing Data Augmentation to Improve Adversarial Robustness TRADES adversarial training on L2 eps=0.5 additional 1M synthetic images in training	91.79	65.44	457.71	2.76	×	WRN-28-10	☐
3	Learning to Generate Noise for Multi-Attack Robustness train with Linf eps=8/255, L2 eps=128/255, L1 eps=2000/255	81.45	63.54	1494.00	8.84	×	WRN-28-10	☐

Figure 2: Top three entries on our leaderboard for *average* case multiattack robustness on CIFAR-10

Rank	Defense	Clean Acc	CR	SC	% Attacks Seen	Extra data	Architecture	Select
1	Formulating Robustness Against Unforeseen Attacks train with Fast LPA	72.47	3.30	4947.00	11.60	×	ResNet-50	☐
2	Perceptual Adversarial Robustness: Defense Against Unseen Threat Models train with Fast LPA	71.58	3.04	5823.00	11.60	×	ResNet-50	☐
3	Manifold Regularization for Locally Stable Deep Neural Networks standard training with Hamming regularization on activation patterns	72.14	2.47	1.75	0.55	×	ResNet-18	☐

Figure 3: Top three entries on our leaderboard for *worst* case multiattack robustness on CIFAR-10

that by using ResNet-18 accuracies for $\text{acc}^*(P)$, our metrics are also able to capture improvements in multiattack performance due to changes in architecture.

Attack strength function. In Section 3.2, we defined SC in terms of an attack strength function. For our leaderboard, we choose to use the error of a ResNet-18 model trained directly on the attack as the attack strength function. For computing SC as in Definition 3.7, we use $\alpha = 3\%$.

4.2. Leaderboard

We provide 2 leaderboards for the CIFAR-10 dataset, one for average case performance, which ranks defenses based on $\text{CR}_{\text{ind-avg}}$, and one for worst case performance, which ranks defenses based on $\text{CR}_{\text{ind-worst}}$. Our leaderboard contains evaluations for 16 pretrained models, all of which use training-based defenses, including techniques for training on unions of ℓ_p norms (Maini et al., 2020; Tramèr & Boneh, 2019; Madaan et al., 2020), training with novel threat models (Laidlaw et al., 2021), regularization based approaches (Jin & Rinard, 2020; Dai et al., 2022), and ℓ_p norm adversarial training (Madry et al., 2018; Zhang et al., 2019; Rebuffi et al., 2021). We include details of the models present on the leaderboard in Appendix D. We note that these models are trained with either ℓ_2 attacks with $\epsilon = 0.5$, ℓ_∞ attacks with $\epsilon = \frac{8}{255}$, LPIPS attacks with $\epsilon = 1$, or the union of $\ell_1, \epsilon = \frac{2000}{255}$, $\ell_2, \epsilon = \frac{128}{255}$ and $\ell_\infty, \epsilon = \frac{8}{255}$ attacks.

We compute ranks based on the set of attacks described in 4.1. We note that none of the models evaluated use all the attacks in our evaluation set, so *all models are evaluated for performance in a partial knowledge or no knowledge setting*. The top 3 entries on each leaderboard are shown in Figure 2 and Figure 3. Our leaderboard site also provides features such as performance visualizations. We discuss these further in Appendix F.

5. Analysis

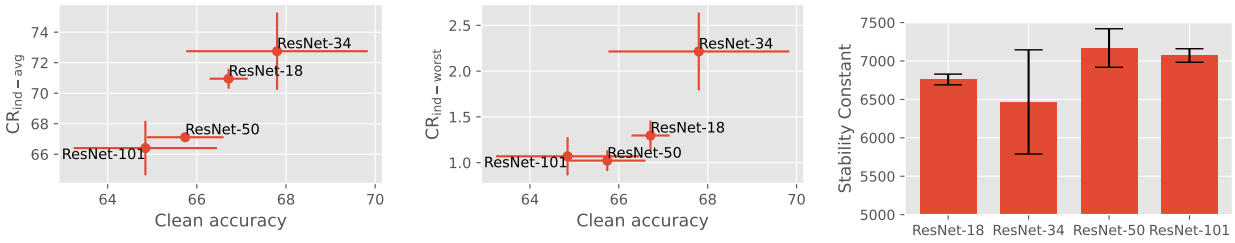
Using our proposed metrics and leaderboard evaluations, we now analyze the performance of existing techniques for multiattack robustness (specifically under a partial or no knowledge setting). Additionally, since some entries on our leaderboard utilize larger architecture size and additional training data, we separately study the impact of these factors on CR and stability to provide deeper insights as to how these design choices influence multiattack robustness.

5.1. Evaluating existing techniques for robustness against multiple perturbations

To understand the performance of existing defenses for multiattack robustness, we plot clean accuracy, $\text{CR}_{\text{ind-avg}}$, and $\text{CR}_{\text{ind-worst}}$ across defenses in Figure 1.

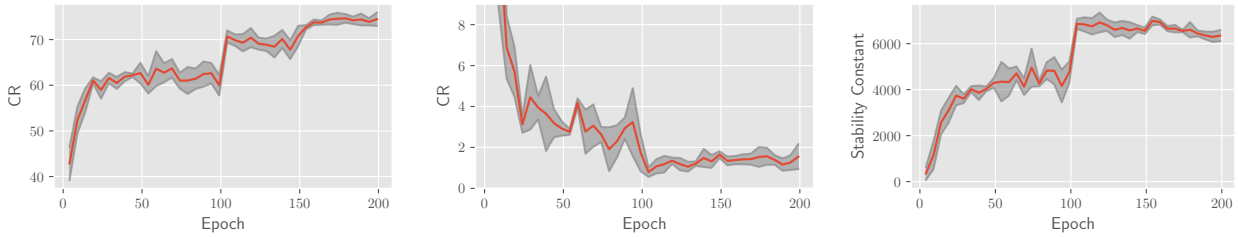
Average case vs worst case multiattack performance. Interestingly, we find that while many defenses can reach high values of $\text{CR}_{\text{ind-avg}}$ (the highest being 67.76), the scores for $\text{CR}_{\text{ind-worst}}$ are much lower (the highest being 3.30). This suggests that for all existing defenses, there are some attacks (which may lie outside of the learner knowledge set) that can significantly reduce the accuracy of the defended model. *Thus, for the task of robustness against the worst-case imperceptible attack, designing defenses that a robust to multiple attacks is a significant open problem for the research community.* This trade-off between clean accuracy and robust accuracy has been noted in prior works (Tsipras et al., 2019; Zhang et al., 2019; Tramèr & Boneh, 2019).

Clean accuracy vs average case multiattack performance. From Figure 1a, we find that some existing defenses achieve



(a) Clean accuracy vs $CR_{ind-avg}$ (b) Clean accuracy vs $CR_{ind-worst}$ (c) SC

Figure 4: **Impact of architecture size.** Figures (a) and (b): Clean accuracy vs CR for models trained with PAT (Laidlaw et al., 2021) (LPIPS threat model). Results are averaged over 3 trials and error bars are shown. Higher values of CR indicate better performance. Figure (c): SC computed for models of each architecture. Lower SC indicates better performance.



(a) $CR_{ind-avg}$ (b) $CR_{ind-worst}$ (c) SC

Figure 5: **Impact of number of training epochs.** CR and SC over epoch for models trained using PAT (Laidlaw et al., 2021) (LPIPS threat model). The red line indicates the average over 3 runs while the grey band highlights indicate 1 standard deviation from the mean. Higher values of CR and lower values of SC indicate better performance.

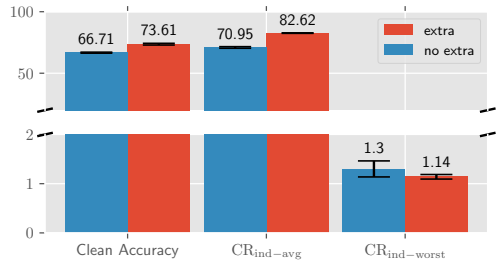
both high clean accuracy and high $CR_{ind-avg}$. Interestingly, we find that the 2 best models in terms of $CR_{ind-avg}$ and clean accuracy ((Madaan et al., 2020) with robust self-training and (Rebuffi et al., 2021) with ℓ_2 threat model) both incorporate additional training data which suggests that additional training data may improve average performance over attacks tested. Overall, we find that $CR_{ind-avg}$ are uncorrelated; for example, the rank 4, 5, and 6 models in terms of $CR_{ind-avg}$ (models using LPIPS threat model (Laidlaw et al., 2021; Dai et al., 2022) and no knowledge (Jin & Rinard, 2020)) have the lowest clean accuracies out of all defenses present on the leaderboard.

Clean accuracy vs worst-case multiattack performance. From Figure 1b, we observe that the models with highest $CR_{ind-worst}$ also have the lowest clean accuracy, which differs from trends observed for $CR_{ind-avg}$. We note that the models achieving the top 3 $CR_{ind-worst}$ scores are in fact the rank 4, 5, and 6 models in terms of $CR_{ind-avg}$. The state of current defenses in $CR_{ind-worst}$ also suggests that *there may be some trade-off between worst-case multiattack performance and clean accuracy*.

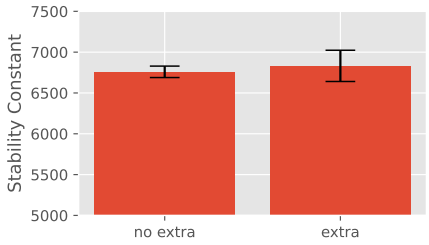
CR across individual attacks. In Figure 1c, we plot single-$CR_{ind-avg}$ (CR computed across individual attack types) averaged over all 16 defenses. We find that out of all attack types, attacks that spatially perturb pixels (elastic attacks and StAdv attacks) are generally the most challenging to defend against. In fact, the best performing model on elastic attacks can only achieve single-$CR_{ind-avg}$ score of 38.48 for elastic attacks. Meanwhile, for StAdv attacks, the highest single-$CR_{ind-avg}$ score is 50.35. We note that these scores are not obtained by the top 3 models ranked by $CR_{ind-avg}$, and are obtained by rank 7 ((Rebuffi et al., 2021) with ℓ_∞ threat model) and rank 6 ((Laidlaw et al., 2021)) respectively. This suggests that *designing defenses that have improved performance on elastic and StAdv attacks can improve the state of current defenses for multiattack robustness*.

5.2. Understanding the impact of architecture size, additional training data, and early stopping on multiattack robustness

While all evaluated models on our leaderboard use training-based defenses which can be applied to any architecture and training dataset, the entries differ in choice of architecture, use of additional training data, and number of training epochs used. To investigate the impact of how these factors influence multiattack robustness, we train ResNet models using adversarial training with 3 different threat models (LPIPS with radius 0.5 (Laidlaw et al., 2021), ℓ_∞ with



(a) Clean Accuracy and CR



(b) SC

Figure 6: **Impact of additional training data.** Figure (a): Clean accuracy and CR for ResNet-18 models trained using PAT (Laidlaw et al., 2021) (LPIPS threat model). Higher CR indicates better performance. Results are averaged over 3 trials and error bars are shown. Figure (c): SC computed for models with and without additional training data. Lower SC indicates better performance.

radius $\frac{8}{255}$ and ℓ_2 with radius 0.5) and analyze $\text{CR}_{\text{ind-avg}}$, $\text{CR}_{\text{ind-worst}}$, and SC of trained models. We present results for LPIPS threat model in this section and provide corresponding analysis for ℓ_∞ and ℓ_2 threat models in Appendix H.3.1. Details about experimental setup are available in Appendix G. We note that to reduce computational cost, we evaluate ℓ_∞ and ℓ_2 robustness using PGD and ℓ_1 robustness using APGD-CE (Croce & Hein, 2020b) instead of running APGD-T and FAB-T attacks as done for leaderboard evaluations, so CR scores present in this section are not directly comparable to those on the leaderboard.

Impact of architecture size. We present results on the impact of architecture size on CR and SC in Figure 4. We find that out of the architectures tested (ResNet-18, ResNet-34, ResNet-50, ResNet-101), smaller architectures (ResNet-18 and ResNet-34) generally have higher clean accuracy and higher CR compared to ResNet-50 and ResNet-101. While previous studies (Gowal et al., 2020) demonstrated that larger architecture can improve robust performance for ℓ_p robustness, we find that this is not always the case for multiattack robustness (even with ℓ_p training as shown in Appendix H.3.1). The higher CR values suggest that these *smaller models have better generalization to unseen attacks while larger models are more likely to overfit to seen attack types*. We find that SC is also on average lower for smaller architectures which indicates that *smaller architectures have less change in robust accuracy across attack types*.

Impact of number of epochs. We present results on the impact of number of training epochs on CR and SC in Figure 5. We observe that while $\text{CR}_{\text{ind-avg}}$ generally increases over training epochs, $\text{CR}_{\text{ind-worst}}$ decreases over epochs, indicating that average robustness increases, but worst-case robustness does not. *This suggests that while more training improves average performance across the set of tested attacks, there may be a few attacks in this set for which more training degrades performance.* When we investigate this further, we find that for all attacks except for elastic attacks, $\text{CR}_{\text{ind-worst}}$ increases over epochs. In Appendix H.2.2, we plot the $\text{CR}_{\text{ind-worst}}$ for each attack type and investigate the impact of including elastic attacks in training. The trend in worst-case robustness is also reflected by Figure 5c which shows that SC increases over training epochs, meaning there is a large drop in robustness when evaluated on unseen attacks.

Impact of additional training data. We now investigate the impact of using additional (synthetic) training data. Specifically, we incorporate the 1M DDPM samples from (Gowal et al., 2021) for CIFAR-10 into training. We present results on the impact of additional data on CR and SC in Figure 6. From Figure 6a, we find that for $\text{CR}_{\text{ind-avg}}$, using additional data significantly improves clean accuracy and CR scores, suggesting that the *extra training data can improve average robustness across attacks*. In fact, we achieve SOTA $\text{CR}_{\text{ind-avg}}$ (69.14) when compared to other models on the leaderboard. For worst-case performance ($\text{CR}_{\text{ind-worst}}$), we find that CR with and without extra data is comparable. This suggests that while on average extra data helps, extra data does not uniformly improve performance across all attacks. In Appendix H.2.3, we plot the impact of additional data on $\text{CR}_{\text{ind-worst}}$ for each attack type. Similar to the overall trend for $\text{CR}_{\text{ind-worst}}$, we find that extra training data does not have much impact on stability; Figure 6b shows that the SCs are comparable with and without extra data.

6. Limitations, Discussion, and Conclusion

The need for a benchmark is imperative to better understand and standardize the progress in multiattack robustness. In our benchmark, we introduce new metrics (CR and SC) and a leaderboard which ranks models based on a set of 180 attacks using these metrics. Currently, our leaderboard contains 16 models, and as new defenses for multiattack robustness are proposed, we plan to update it with new defenses. Additionally, as new attack types and stronger attacks are introduced, we plan to incorporate these into our evaluation pipeline.

One challenge with our benchmark is the runtime of evaluation, which makes it very computationally expensive to eval-

uate large architectures and large-scale image datasets, such as ImageNet. Future improvements in attack efficiency can improve the scalability of our evaluation pipeline. Currently, our leaderboard only contains evaluations for CIFAR-10; the future, we hope to include additional leaderboards for other image datasets.

Our benchmark and analyses highlight the weaknesses of current defenses on the task of worst-case multiattack robustness; in particular, we find that no defense can outperform random guessing. In addition, we demonstrate that trends for single (known) attack robustness do not necessarily hold for the multiattack robustness. We hope that our benchmark inspires future research in multiattack robustness.

Acknowledgements

This work was supported in part by the National Science Foundation under grants CNS-2131938, the ARL's Army Artificial Intelligence Innovation Institute (A2I2), Schmidt DataX award, and Princeton E-ffiliates Award. This material is also based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-2039656. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., and Duchi, J. C. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019.

Croce, F. and Hein, M. Provable robustness against all adversarial l_p-perturbations for $p\geq 1$. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020a. URL https://openreview.net/forum?id=rklk_ySYPB.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020b.

Croce, F. and Hein, M. Mind the box: l_1-apgd for sparse adversarial attacks on image classifiers. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2201–2211. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/croce21a.html.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Dai, S., Mahloujifar, S., and Mittal, P. Formulating robustness against unforeseen attacks. *arXiv preprint arXiv:2204.13779*, 2022.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.

Hsiung, L., Tsai, Y.-Y., Chen, P.-Y., and Ho, T.-Y. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. *arXiv preprint arXiv:2202.04235*, 2022a.

Hsiung, L., Tsai, Y.-Y., Chen, P.-Y., and Ho, T.-Y. CARBEN: Composite Adversarial Robustness Benchmark. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. International Joint Conferences on Artificial Intelligence Organization, July 2022b.

Huang, H., Wang, Y., Erfani, S., Gu, Q., Bailey, J., and Ma, X. Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 34:5545–5559, 2021.

Jin, C. and Rinard, M. Manifold regularization for locally stable deep neural networks. *arXiv preprint arXiv:2003.04286*, 2020.

Kang, D., Sun, Y., Hendrycks, D., Brown, T., and Steinhardt, J. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

Laidlaw, C. and Feizi, S. Functional adversarial attacks. *Advances in neural information processing systems*, 32, 2019.

Laidlaw, C., Singla, S., and Feizi, S. Perceptual adversarial robustness: Defense against unseen threat models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=dFwBosAcJkN.

Madaan, D., Shin, J., and Hwang, S. J. Learning to generate noise for robustness against multiple perturbations. *CoRR*, abs/2006.12135, 2020. URL https://arxiv.org/abs/2006.12135.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6640–6650. PMLR, 2020. URL http://proceedings.mlr.press/v119/maini20a.html.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.

Rice, L., Wong, E., and Kolter, Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., and Mittal, P. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1904.13000.

Tsai, Y.-Y., Hsiung, L., Chen, P.-Y., and Ho, T.-Y. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations, 2022.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, number 2019, 2019.

Wu, B., Chen, J., Cai, D., He, X., and Gu, Q. Do wider neural networks really help adversarial robustness? *Advances in Neural Information Processing Systems*, 34: 7054–7067, 2021.

Xiao, C., Zhu, J., Li, B., He, W., Liu, M., and Song, D. Spatially transformed adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19p.html.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00068. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

A. Categorization of existing defenses against multiple attacks

We now categorize existing defenses for multiattack robustness into the different knowledge settings explored and provide a brief description of each defense.

A.1. Full Knowledge

The works under the full knowledge category generally study the problem of achieving robustness against unions of attacks (typically ℓ_p norms).

AVG and MAX training (Tramèr & Boneh, 2019) Tramèr & Boneh (2019) propose 2 methods for adversarial training with a set of different attack types. In both methods, for every training example, the learner generates adversarial examples with respect to all attacks in the set. With AVG training, the loss used for backpropagation is taken to be the average of losses across this set of adversarial examples, while with MAX training, the loss is taken to be the maximum loss over the set of adversarial examples. Since these techniques are targeted towards robustness against the same set of attacks as used during training, this defense is designed as a defense with full knowledge.

Multiple steepest descent (MSD) (Maini et al., 2020) Maini et al. (2020) improve upon the MAX training algorithm from Tramèr & Boneh (2019) specifically for robustness against union of ℓ_1, ℓ_∞, and ℓ_2 norm attacks. For these norms, adversarial examples are generally obtained through multiple steps of gradient-based optimization (PGD). Instead of applying 3 rounds of PGD to obtain ℓ_1, ℓ_∞, and ℓ_2 attacks and then taking the maximum loss for backpropagation, MSD unrolls the PGD steps and at each step of PGD chooses the the PGD update with respect to the norm that maximizes the loss after the update. This technique also falls under the category of full knowledge because during evaluation, the models are tested on the same attacks as used during the training process.

Stochastic adversarial training (SAT) (Madaan et al., 2020) Madaan et al. (2020) propose stochastic adversarial training (SAT), where to achieve robustness against a set of attacks, for each input an attack from that set is chosen at random. The authors then combine this with a regularization which enforces similar distributions of prediction probabilities for clean inputs, adversarial examples, and noisy inputs, with noise generated through their proposed meta-noise generator (MNG). Since this technique also sees all the attack types in training as during evaluation, this defense also falls under the full knowledge setting.

A.2. Partial Knowledge

In practice, there can be mismatch between the attacks used by the learner and attacks used during test-time by the adversary, which motivates studying the partial knowledge setting. Recently, two works have begun investigating defending in the partial knowledge attacks. In general, the problem of defending against unforeseen attacks falls under this category.

Perceptual adversarial training (Laidlaw et al., 2021) Laidlaw et al. (2021) propose a adversarial training technique called perceptual adversarial training (PAT) which uses attacks that are based on LPIPS metric. LPIPS (Zhang et al., 2018) is a distance metric that is based on distances between feature maps when images are passed through a trained neural network (ie. AlexNet) and has been demonstrated to be more perceptually aligned than ℓ_p distance metrics. Laidlaw et al. (2021) show that models trained using PAT can exhibit nontrivial robustness against ℓ_p attacks, ReColor attacks (Laidlaw & Feizi, 2019), and StAdv attacks (Xiao et al., 2018). Since these attacks were not used during training, this defense falls under the partial knowledge setting.

Variation regularization (Dai et al., 2022) Dai et al. (2022) propose a regularization technique called variation regularization for reducing drop in robust accuracy to unseen threat models. This regularization technique any two perturbed inputs from the train-time threat model have similar predicted logits. They, then combine this regularization method with adversarial training methods (such as PAT and PGD adversarial training) and evaluate the regularized model on attacks such as ℓ_p attacks, ReColor attacks (Laidlaw & Feizi, 2019), and StAdv attacks (Xiao et al., 2018), which are outside of the train-time threat model. Thus, this technique falls under the category of partial knowledge.

A.3. No Knowledge

Manifold regularization (Jin & Rinard, 2020) Currently, to the best of our knowledge Jin & Rinard (2020) is the only defense which utilizes no knowledge of the test-time threat model. Jin & Rinard (2020) propose using two regularization terms with standard training, one which reduces the hamming distance of activation patterns between perturbed images

and one that reduces the ℓ_2 Lipschitz constant of the network. The perturbations used for regularization are not adversarial, instead they are random. They show that their technique is able to achieve nontrivial ℓ_∞, ℓ_2, and LPIPS robustness. Since adversarial examples are not used during training, this defense falls under the no knowledge category.

B. Comparison to Existing Evaluation Methods

Previous works in multiattack robustness generally utilize 4 different approaches for evaluating robustness. In this section, we discuss these techniques and compare our evaluation method to these techniques.

Accuracy on individual attack types Most works in multiattack robustness report robust accuracy on individual attacks at some chosen attack strength ϵ. For example, works on adversarial training with multiple ℓ_p norms (Madaan et al., 2020; Maini et al., 2020; Tramèr & Boneh, 2019) report robust accuracies for the ℓ_p attacks used during training. While this is approach provides the most information about robustness on individual attacks, typically these numbers are reported for only a few attack types at a single attack strength per attack type. In our work, we evaluate 9 different attack types with 20 levels of attack strength leading to a larger scope in evaluation. In our performance visualizations (See Appendix F), we also allow users to see the CR computed across individual attack types so that users are still able to understand relative performance across each attack type.

Accuracy on the union of different attacks Another commonly reported value is the accuracy across the union of different attack types, which is obtained by considering an image incorrectly classified if any of the attacks in the attack set succeed. While this metric is a good approximation of worst case robustness, this metric is commonly reported for only a few attack types at a single attack strength per attack type. This metric is also does not take into account the inherent difficulty of each attack which can bias scores. For example, consider a setting where one attack P in the evaluation set is inherently more difficult than the rest and the best model for this attack can do no better than random guessing. In this case, we would always expect the union accuracy to be highly biased by P and always have value less than $\frac{1}{K}$ where K is the number of classes. Our metric $\text{CR}_{\text{ind-worst}}$ addresses this bias by weighting the robust accuracy of the defense by $\frac{1}{\text{acc}^*(P)}$.

Average accuracy across attacks Another value reported by papers in multiattack robustness is average accuracy across attacks. For example, Laidlaw et al. (2021) report average accuracy across unseen attacks to demonstrate improved robustness against unseen attacks. Similar to union accuracy, this metric can also be biased by attack difficulty. Our metric $\text{CR}_{\text{ind-avg}}$ addresses this bias by weighting the robust accuracy of the defense by $\frac{1}{\text{acc}^*(P)}$.

mUAR metric (Kang et al., 2019) Kang et al. (2019) introduce a metric called mUAR for evaluating robustness against unseen attacks. Specifically, this value is defined as follows:

Definition B.1 (mUAR (Kang et al., 2019))**.** Let K be a set of different attack types P. Let $\text{acc}(h, P, \epsilon)$ denotes the robust accuracy of defended model h using attack P with attack strength ϵ. Let $\text{acc}^*(P, \epsilon)$ denote the best accuracy obtainable from a model in $\mathcal{H}$. For each $P_i \in K, i \in \{1...|K|\}$, let $\mathcal{E}_i$ be a corresponding set of attack strengths. Then, for a model h,

$$\text{UAR}(h, P, \mathcal{E}) = 100 \times \frac{\sum_{\epsilon \in \mathcal{E}} \text{acc}(h, P, \epsilon)}{\sum_{\epsilon \in \mathcal{E}} \text{acc}^*(P, \epsilon)} \tag{5}$$

$$\text{mUAR}(h) = \frac{1}{|K|} \sum_{i=1}^{|K|} \text{UAR}(h, P_i, \mathcal{E}_i) \tag{6}$$

From the definition of UAR, for a single attack type the aggregated robust accuracies across attack strengths ϵ are weighted by the aggregate best accuracy attainable, which addresses the problem of bias from evaluated across different values of ϵ. However, when considering multiple attacks mUAR weights the scores of each attack equally, so this score can be still be biased by the difficulties of each attack type. In comparison, our CR metrics are weighted across different attack types as well. We also note that by using a different definition of multiattack error and using a single attack type in our evaluation set, we can obtain the UAR metric from CR (see Appendix E for more discussion).

CARBEN (Hsiung et al., 2022b) (Hsiung et al., 2022b) propose a benchmark for measuring compositional robustness called CARBEN. In CARBEN evaluates models by optimizing the attack order of a set of attacks at a single attack strength (specifically ℓ_∞, hue, saturation, rotation, brightness, contrast) and reporting the robust accuracy of the model after performing that sequence of attacks. In general, we find that hue, saturation, rotation, brightness, and contrast attacks are much weaker than existing (and less perceptible) attacks such as StAdv and ReColor, so the accuracies from the CARBEN benchmark does not reflect multiattack robustness well. Additionally, we evaluate on multiple attack strengths for each

attack and use CR metrics due to potential bias from using accuracy.

C. Description of Attacks Used and Evaluation Procedure

In this section, we describe in more depth the attacks used by the benchmark and evaluation procedure for computing CR and stability scores. We include samples of each attack at the strengths evaluated in Figure 7

C.1. Attack descriptions

ℓ_p attacks The most commonly studied form of robustness is robustness to ℓ_p attacks, mainly ℓ_1, ℓ_2, and ℓ_∞ attacks. AutoAttack (Croce & Hein, 2020b) is a commonly used package for evaluating the robustness which includes a combination of 3 white box attacks (APGD-CE, APGD-T, and FAB-T) and a black box attack (Square). Since we evaluate at 20 different attack strengths for each attack type, we evaluate with only the APGD-T and FAB-T attacks to reduce evaluation time. For ℓ_1 attacks, we evaluate with attack strength $\epsilon \in (0, 30]$ in increments of 1.5. For ℓ_2 attacks, we evaluate with $\epsilon \in (0, 2.5]$ in increments of 0.125. For ℓ_∞ attacks, we evaluate with $\epsilon \in (0, 0.1)$ in increments of 0.005.

Spatial transformation attacks For the leaderboard, we measure robustness to spatial transformations using StAdv attack (Xiao et al., 2018). This attack generates adversarial examples by optimizing for a per pixel flow field f, where f_i corresponds to the displacement vector of the ith pixel of the image. This flow field is obtained by solving $\arg\min_f \ell_{\text{adv}}(x, f) + \tau\ell_{\text{flow}}(f)$ where ℓ_{adv} is the CW objective (Carlini & Wagner, 2017) and ℓ_{flow} is a regularization term that controls the smoothness of the change. τ is a hyperparameter controlling regularization strength. For StAdv attacks, we evaluate with $\epsilon \in (0, 0.1]$ in increments of 0.005 and set $\tau = 0.0025/\epsilon$.

Outside of StAdv attacks, we also allow users to see robust accuracies for attacks that apply global spatial transformations including affine warp and perspective warp (though these are not included in the leaderboard ranking as the attacks are much easier than StAdv). An affine warp is a transformation captured by a matrix of the form $M_{\text{affine}} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ 0 & 0 & 1 \end{pmatrix}$ and the pixel coordinates of the resulting image is obtained by multiplying M_{affine} to each pixel coordinate vector $(x, y, 1)^T$. Affine transformations capture translations, rotations, scaling, and shear. To generate adversarial affine transformations, we use PGD to optimize over M_{affine} and apply an ℓ_∞ constraint to M_{affine}. For affine attacks, we use ℓ_∞ bounds in range (0, 0.1) with increments of 0.005. Perspective warps capture more transformations than affine warps and can be parameterized by a matrix $M_{\text{perspective}} = \begin{pmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & 1 \end{pmatrix}$. We apply the same method to optimize over $M_{\text{perspective}}$ and apply an ℓ_∞ constraint. For perspective attacks, we use ℓ_∞ bounds in range (0, 0.002) with increments of 0.0001.

Color shift attacks For the leaderboard, we measure robustness to color shifts via ReColor attacks (Laidlaw & Feizi, 2019). The approach to generating ReColor attacks is similar to the objective of StAdv attacks: $\arg\min_f \ell_{\text{adv}}(x, f) + \tau\ell_{\text{flow}}(f)$, where f now models a function that maps between colors and $\ell_{\text{flow}}(f)$ is a regularization term that ensures that neighboring pixels will have colors changed in a similar manner. For ReColor attacks, we evaluate with $\epsilon \in (0, 0.1]$ in increments of 0.005 and set $\tau = 0.0036/\epsilon$.

We also allow users to see evaluations for global color changes including hue shifts, brightness changes, contrast changes, and saturation changes (but these scores are not used for computing leaderboard ranking as they are much weaker attacks than ReColor). All of these color changes can be parameterized by a single scalar parameter, and we use PGD to optimize this parameter as in Tsai et al. (2022). For hue and saturation attacks, we consider changes in the parameter from (0, 0.5] with increments of 0.025. For brightness, we consider changes from (0, 0.3] with increments of 0.015, and for contrast, we consider changes from (0, 0.5] with increments of 0.025.

UAR attacks (Kang et al., 2019) Kang et al. (2019) propose a set of attacks for evaluating unforeseen robustness including attacks such as elastic attacks, ℓ_1 JPEG attacks, ℓ_∞ JPEG attacks, snow, fog, and Gabor attacks. Of these attacks, elastic attacks, ℓ_1 JPEG attacks, and ℓ_∞ JPEG attacks are targeted towards the CIFAR-10 dataset, so we incorporate these attacks into leaderboard ranking. Elastic attacks are a spatial attack based off of StAdv where the flow field f is obtained by smoothing a vector W by a Gaussian kernel and optimizing over W under the constraint that $||W||_\infty \leq \epsilon$. For elastic attacks, we consider $\epsilon \in [0, 1)$ in increments of 0.05. ℓ_1 (or ℓ_∞) JPEG attacks optimize for ℓ_1 (or ℓ_∞) bounded adversarial examples in the JPEG-encoded space of compressed images. For ℓ_1 JPEG attacks, we consider ℓ_1 bounds in range (0, 20] in

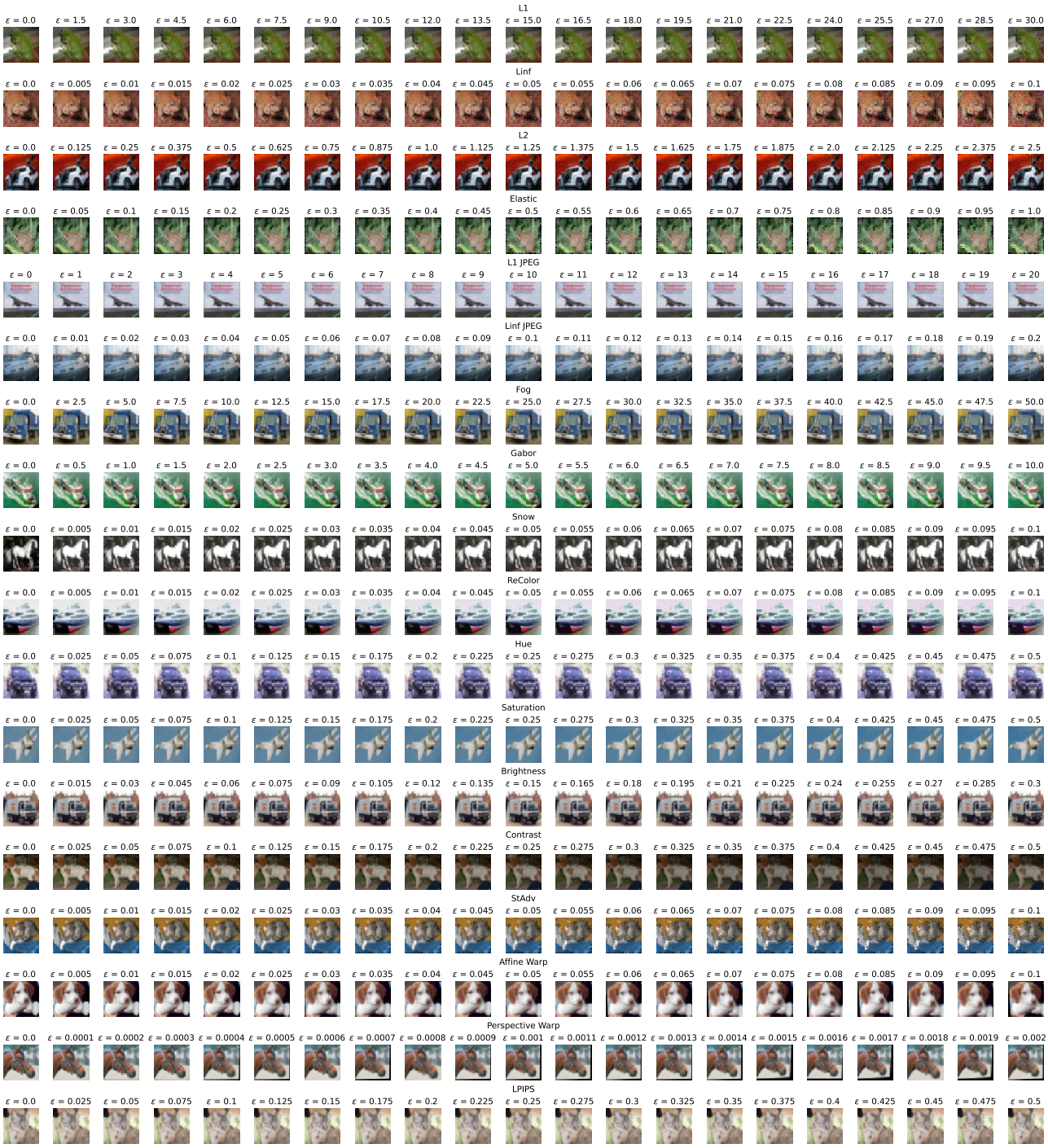


Figure 7: Samples of all attacks at each attack strength ϵ used in evaluation.

increments of 1. For ℓ_∞ JPEG attacks, we consider ℓ_∞ bounds in range (0, 0.2] in increments of 0.01.

While they are not included in leaderboard ranking, we allow users to see the scores for snow, fog, and Gabor attacks on the leaderboard site. For snow attacks, we provide evaluations for attack strengths in range (0, 0.1) in increments of 0.005. For fog attacks, we provide evaluations for attack strengths in range (0, 50) in increments of 2.5. For gabor attacks, we provide evaluations for attack strengths in range (0, 10) in increments of 0.5.

Bounded LPIPS attacks Laidlaw et al. (2021) LPIPS distance (Zhang et al., 2018), is a distance metric based on distances between feature maps of trained models (ie. AlexNet) and has been shown to be more perceptually aligned than ℓ_p metrics. Laidlaw et al. (2021) introduce 2 attacks, perceptual PGD (PPGD) and Lagrange perceptual attack (LPA) based on this distance. PPGD optimizes for adversarial examples in a way that is analogous to PGD: at each iteration, an optimal first order step is taken and then projected so that it lies in the LPIPS bound. LPA moves the distance constraint into the objective function via Lagrangian relaxation. For evaluation, we measure robustness against LPIPS attacks by taking the accuracy against the union of PPGD and LPA attacks based off of AlexNet architecture. We consider attacks of LPIPS bound in range (0, 0.5] in increments of 0.025.

C.2. Additional evaluation details

In this section, we describe additional details about how we perform metric computations.

Training models for approximating optimal single attack accuracy To approximate optimal single attack accuracy, for each attack, we train 3 ResNet-18 models using adversarial training. We also train a set of 3 models with standard training (no attack). For ℓ_1 threat model, we train using adversarial examples generated via APGD (Croce & Hein, 2021). For ℓ_2 and ℓ_∞ threat models, we use PGD-adversarial training. For LPIPS threat model, we use perceptual adversarial training (PAT) which uses a fast approximation to the LPA attack Laidlaw et al. (2021). For all other threat models, we the same attack generation method during training as used for evaluation as described in the previous section. For all threat models (with the exception of ℓ_1 threat model which uses settings from Croce & Hein (2021) and UAR attacks which use default settings from Kang et al. (2019)), we use 20 iterations to find adversarial examples with step size $\epsilon/18$. We train all models with batch size of 256 for 100 epochs and evaluate the model saved at the epoch which achieves highest robust accuracy on the test set. We train models using SGD with initial learning rate of 0.1. Learning rate drops to 0.01 after half of the training epochs and drops to 0.001 after 3/4 of the training epochs.

Robust accuracy evaluation For evaluating robust accuracy, with the exception of ℓ_p attacks and UAR attacks which use default evaluation setups from Croce & Hein (2020b) and Kang et al. (2019) respectively, we use 20 iterations to optimize over adversarial examples for non-LPIPS threat models and 40 iterations to optimize over adversarial examples for LPIPS threat model. To obtain robust accuracies at multiple epsilon per attack type, we perform robustness evaluations in a binary search manner to find the smallest perturbation size at which the model misclassifies each image. We aggregate this information across the entire test set to find the robust accuracy at each value of epsilon.

Metric computation For computing $\text{CR}_{\text{ind-avg}}$ and $\text{CR}_{\text{ind-worst}}$, we follow Definition 3.5 and take K to be the set of 180 attacks described in C.1 (9 attacks at 20 different values of ϵ each) with 1 additional attack representing no attacker ($\epsilon = 0$). For $\text{CR}_{\text{ind-avg}}$, we assume that the distribution over all attacks is uniform.

For computing stability constant, we consider K_{learner} to include attack strength ϵ from 0 (no attack) to the ϵ that is used by the defense that fall within the 20 values of ϵ that we used for evaluating robust accuracy. For example, if a defense uses ℓ_2 threat model with strength 0.5, and in our evaluation procedure for ℓ_2, we evaluate with $\epsilon \in (0, 2.5]$ in increments of 0.125, we would consider K_{learner} to contain no attack ($\epsilon = 2$) and ℓ_2 attacks with $\epsilon \in \{0.125, 0.25, 0.375, 0.5\}$. We then follow the equation in Definition 3.7 with $\alpha = 3\%$. We note that we found that both reducing α to 1% and increasing α generally does not influence the SC of defended models. The only defense whose SC changed as a result of changing α was (Jin & Rinard, 2020) for which K_{learner} does not contain any attacks (outside of $\epsilon = 0$), for which decreasing α to 1% reduces stability constant to 0 due to few acc^* values that lie in the viscinity of standard training clean accuracy. We choose 3% to ensure that Jin & Rinard (2020) and other future defenses which may also be based on standard training will still have a value for SC for comparison.

D. Defenses Present on Leaderboard

Overall, we evaluate a total of 16 different models. All defenses use one of the follow training threat models: no attack (standard training), ℓ_2 with $\epsilon = \frac{128}{255}$, ℓ_∞ with $\epsilon = \frac{8}{255}$, (AlexNet) LPIPS with $\epsilon = 1$, a combination of ℓ_1 with $\epsilon = \frac{2000}{255}$, ℓ_2 with $\epsilon = \frac{128}{255}$, and ℓ_∞ with $\epsilon = \frac{8}{255}$. We describe the entries present on our leaderboard below:

- Single attack adversarial training approaches: Since adversarial training with ℓ_2 and ℓ_∞ is commonly studied, we include 2 entries (one with ℓ_2 attacks, and one with ℓ_∞ attacks) for PGD adversarial training (Madry et al., 2018) and 2 entries for TRADES adversarial training (Zhang et al., 2019) (one with ℓ_2 attacks, and one with ℓ_∞ attacks). All 4 of these entries use ResNet-18 architecture. Prior works have improved on the performance of ℓ_2 and ℓ_∞ through the use of additional synthetic data. One of these approaches is (Rebuffi et al., 2021) which is currently the top performing approach on RobustBench (Croce et al., 2020). We include entries for two WRN-28-10 trained using (Rebuffi et al., 2021) (one with ℓ_∞ threat model and the other with ℓ_2 threat model). The pretrained models for Rebuffi et al. (2021) models are available through RobustBench.

- Multiple attack adversarial training approaches: We also include entries for multiple defenses trained with a combination of ℓ_1 with $\epsilon = \frac{2000}{255}$, ℓ_2 with $\epsilon = \frac{128}{255}$, and ℓ_∞ with $\epsilon = \frac{8}{255}$. These include 2 models using the training approach from Madaan et al. (2020) (one which uses additional data via robust self-training (Carmon et al., 2019) and one that does not use additional data), 1 model using the AVG approach in Tramèr & Boneh (2019), 1 model using the MAX approach in Tramèr & Boneh (2019), and 1 model using MSD from Maini et al. (2020). These models are available through the code repository for Madaan et al. (2020) here. These models all use WRN-28-10 architecture.

- Variation regularization (Dai et al., 2022): Dai et al. (2022) propose variation regularization which can be applied on top of any train-time threat model. We include 3 leaderboard entries for this defense, one using ℓ_∞ threat model, one using ℓ_2 threat model, and one using LPIPS threat model. The ℓ_∞ and ℓ_2 models both use ResNet-18 architecture while the LPIPS model uses ResNet-50 architecture. These models are available through the code repository for Dai et al. (2022) here.

- Perceptual adversarial training (PAT) (Laidlaw et al., 2021): We include an entry for PAT with the AlexNet-based LPIPS attacks. This model uses ResNet-50 architecture and is available through the code repository for Laidlaw et al. (2021) here.

- Manifold regularization (Jin & Rinard, 2020): Jin & Rinard (2020) propose a regularization technique that can be applied on top of standard training and does not use adversarial examples to compute. We include a leaderboard entry for manifold regularization. This model uses ResNet-18 architecture. The pretrained model is available here.

E. Additional CR Definitions

In the main body of the paper, we focused mainly on using $\text{err}_{\text{multi-ind}}$ as the multiattack error when defining CR. Additionally, we can consider using $\text{err}_{\text{multi-exp}}$ and $\text{err}_{\text{multi-max}}$ which leads to 2 new definitions of CR:

$$\text{CR}_{\text{exp}}(h; K) = 100 \times \frac{\mathbb{E}_{P\sim\mathcal{D}(K)}\text{acc}(h, P)}{\mathbb{E}_{P\sim\mathcal{D}(K)}\text{acc}^*(P)} \tag{7}$$

$$\text{CR}_{\text{max}}(h; K) = 100 \times \frac{\min_{P\in K}\text{acc}(h, P)}{\min_{P\in K}\text{acc}^*(P)} \tag{8}$$

We note that for $\text{CR}_{\text{exp}}(h; K)$, when $\mathcal{D}(K)$ is uniform and K contains only attacks of the same type, we obtain the UAR metric proposed by Kang et al. (2019).

For leaderboard rankings, we opted to use the $\text{err}_{\text{multi-ind}}$ definitions of CR since more clearly compares robust accuracy on each specific attack to the corresponding robust accuracy of the optimal, making these metrics more interpretable, while CR_{exp} and CR_{max} both compare aggregates across all defense accuracies and across all optimal accuracies. We find that CR_{exp} and CR_{max} both lead to higher scores relative to $\text{CR}_{\text{ind-avg}}$ and $\text{CR}_{\text{ind-worst}}$ respectively. We also find that ranking by CR_{exp} maintains the rankings of the top 3 best performing models compared to $\text{CR}_{\text{ind-avg}}$. For CR_{max}, we find that the set of top 3 best performing models stays the same, but the rankings are reversed compared to $\text{CR}_{\text{ind-worst}}$.

F. Additional Leaderboard Features

While the scores present on the leaderboard allow us to easily compare the performance of defended models for multiattack robustness, it is hard to understand failure points of specific defenses by looking at the score alone. To this end, we provide additional features that allow users to have a more in depth understanding of model performance.

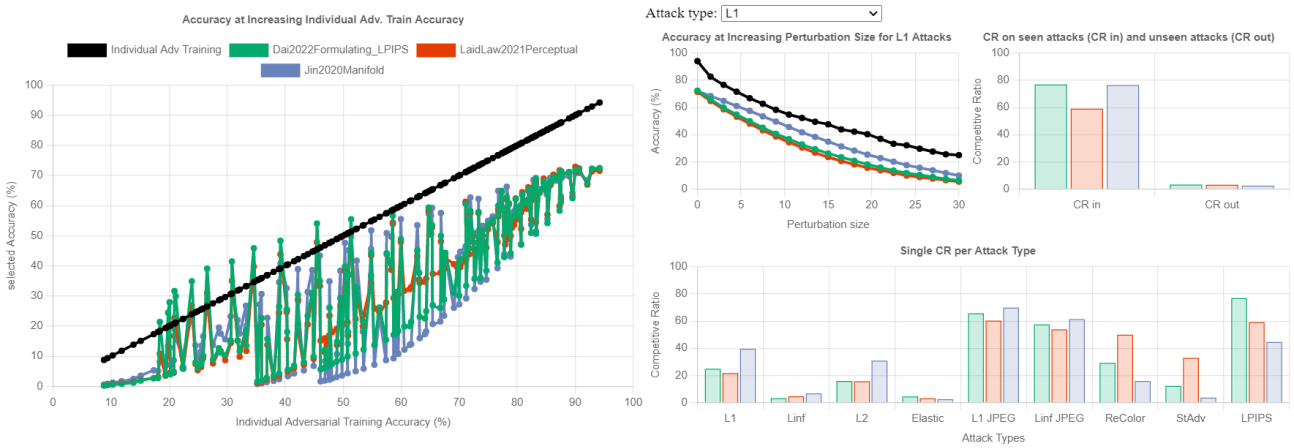


Figure 8: Sample performance visualizations provided on our leaderboard website.

User controls Since some defenses are designed for robustness against a union of specific attack types, we allow users to control the types of attacks used in attack set K used when computing metrics. We note that leaderboard rankings, however, are independent of user selection since the goal of the leaderboard is to reflect performance across a diverse set of imperceptible attacks.

Performance Visualizations While metrics like CR and stability are useful for ranking and comparing the performance of different defenses, it is difficult to understand specific weaknesses of existing defenses, which may make it difficult to improve upon existing techniques. For example, a defense may systematically fail on a particular attack type, but since CR and stability aggregate performance across multiple attack types, they are unable to convey this. To address this, our leaderboard allows users to generate performance visualizations for specific defenses and compare performance visualizations for up to 5 different defenses. We provide 4 types of performance visualizations: (1) a graph of the accuracy of the defense on each tested perturbation type against the accuracy of adversarial training directly on that perturbation type, (2) graphs of robust accuracy across attack strength ϵ for each perturbation type tested, (3) a bar chart of CR-in compared to CR-out (4) a bar chart of single-CR values computed for each perturbation type tested. Examples of these visualizations are shown in Figure 8.

G. Experimental Setup for Adversarially Trained Models

To understand the impact of factors such as architecture size, additional data, and number of training epochs on multiattack performance, we train our own set of models using adversarial training on 3 different threat models: ℓ_2 (with radius $\epsilon = 0.5$), ℓ_∞ (with $\epsilon = \frac{8}{255}$), and LPIPS (with $\epsilon = 0.5$). For ℓ_∞ and ℓ_2 threat models, we train using 20 iterations of PGD with step size $\epsilon/18$. For LPIPS threat model we use PAT with 20 iterations for Fast LPA to find adversarial examples (Laidlaw et al., 2021). For all experiments, we train 3 trials. We train models using SGD with initial learning rate of 0.1. Learning rate drops to 0.01 after half of the training epochs and drops to 0.001 after 3/4 of the training epochs. For all evaluations, we use the same set of attacks as described in Apendix C.1, but for ℓ_∞ and ℓ_2 attacks, we use 10 step PGD to find adversarial examples (with step size $\epsilon/8$), and for ℓ_1 attacks we use APGD (Croce & Hein, 2021).

Architecture experiments We train ResNet-18, ResNet-34, ResNet-50, and ResNet-101 models with batch size 256 for 100 epochs and evaluate at the model saved at the epoch achieving the highest robust accuracy on the test set.

Extra training data experiments We train ResNet-18 models with and without extra 1M (synthetic) training data from Gowal et al. (2021). Models are trained in batches of 150 samples from the original training set and 350 samples from the extra training data. We train for 100 epochs and evaluate at the model saved at the epoch achieving the highest robust

accuracy on the test set.

Training epoch experiments We train ResNet-18 models with batch size of 256 for 200 epochs and save a copy of the model every 5 epochs. We evaluate on this set of saved models.

H. Additional Analysis

H.1. Stability of existing defenses

Since the computation of stability depends on the choice of K_{learner} for fair comparison, we should compare models which use the same threat model during training. In Table 1, we organize defenses present on our leaderboard by train-time threat model and report their corresponding stability constants. We note that the only model missing from Table 1 is model using the manifold regularization defense from Jin & Rinard (2020) as it is the only model that does not use adversarial examples during training.

Defense	Stability Constant
Dai et al. (2022)	1801.00
Rebuffi et al. (2021)	2056.50
Zhang et al. (2019)	2164.00
Madry et al. (2018)	2309.50

(a) ℓ_∞

Defense	Stability Constant
Rebuffi et al. (2021)	457.71
Dai et al. (2022)	904.71
Zhang et al. (2019)	940.29
Madry et al. (2018)	1110.00

(b) ℓ_2

Defense	Stability Constant
Dai et al. (2022)	4947.00
Laidlaw et al. (2021)	5823.00

(c) LPIPS

Defense	Stability Constant
Madaan et al. (2020) (no RST)	1494.00
Maini et al. (2020)	1803.00
Madaan et al. (2020) (RST)	1901.25
Tramèr & Boneh (2019) (MAX)	2145.00
Tramèr & Boneh (2019) (AVG)	2502.00

(d) $\ell_1, \ell_2, \ell_\infty$

Table 1: Stability constants of models present on the leaderboard

We note that of all defenses tested, the defense from Dai et al. (2022) is specifically designed for improving stability (which Dai et al. (2022) refers to as unforeseen generalization gap), and we find that for ℓ_∞, ℓ_2, and LPIPS threat models, the model using Dai et al. (2022) outperforms the corresponding baseline ((Madry et al., 2018) for ℓ_∞ and ℓ_2 norms and (Laidlaw et al., 2021) for LPIPS).

H.2. $\text{CR}_{\text{ind-worst}}$ per attack type analysis for LPIPS trained models

In this section, we present computed $\text{CR}_{\text{ind-worst}}$ values across individual attack types for LPIPS trained models in Section 5.2.

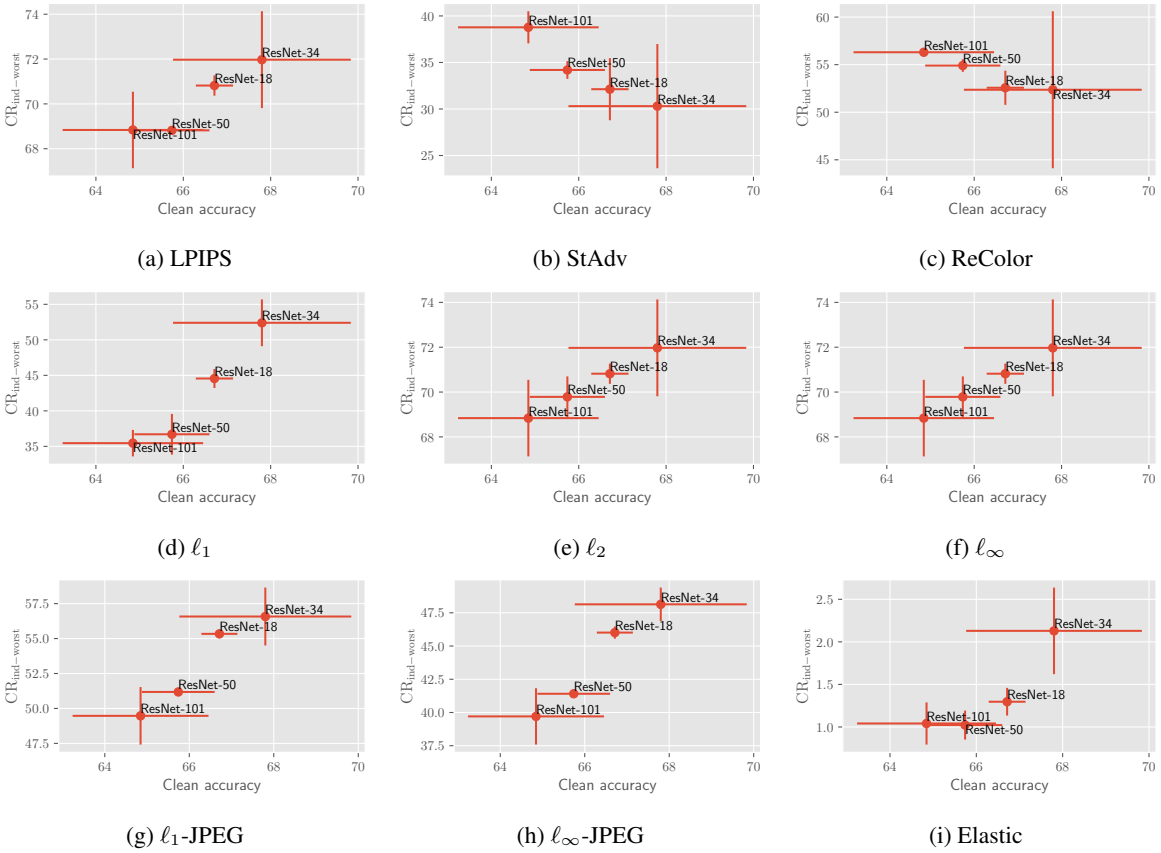
H.2.1. Impact of architecture size

In Figure 9, we plot the impact of architecture size on $\text{CR}_{\text{ind-worst}}$ for each attack type. For StAdv and ReColor attacks, we find that $\text{CR}_{\text{ind-worst}}$ seems inversely correlated with accuracy and larger architectures tend to have higher $\text{CR}_{\text{ind-worst}}$ for those threat models. For all other threat models, we observe that smaller architectures have better performance, which matches our observations in Section 5.2.

H.2.2. Impact of number of training epochs

In Figure 10, we plot the impact of number of training epochs on $\text{CR}_{\text{ind-worst}}$ per attack. We find that for LPIPS threat model (which is used during training), after about 100 epochs, additional training decreases CR on LPIPS attacks. This suggests that after 100 epochs, the model starts to overfit on the training dataset. For other threat models (except elastic attack), we find that CR is generally the highest at the last epoch of trining. For elastic attacks, we find that CR drops during training.

To see if incorporating elastic attacks into training changes the observed trend, we also incorporate elastic attacks with



(a) LPIPS (b) StAdv (c) ReColor

(d) ℓ_1 (e) ℓ_2 (f) ℓ_∞

(g) ℓ_1-JPEG (h) ℓ_∞-JPEG (i) Elastic

Figure 9: Impact of architecture size on $\text{CR}_{\text{ind-worst}}$ per attack type for models trained using LPIPS threat model with $\epsilon = 0.5$ (via FastLPA training (Laidlaw et al., 2021))

$\epsilon = 0.7$ into training (along with the original LPIPS threat model). We train using the maximum of elastic and LPIPS losses and provide plots in Figure 11. Interestingly, we do not observe much difference in training curves between including elastic into training and not including elastic in training.

H.2.3. Impact of extra data

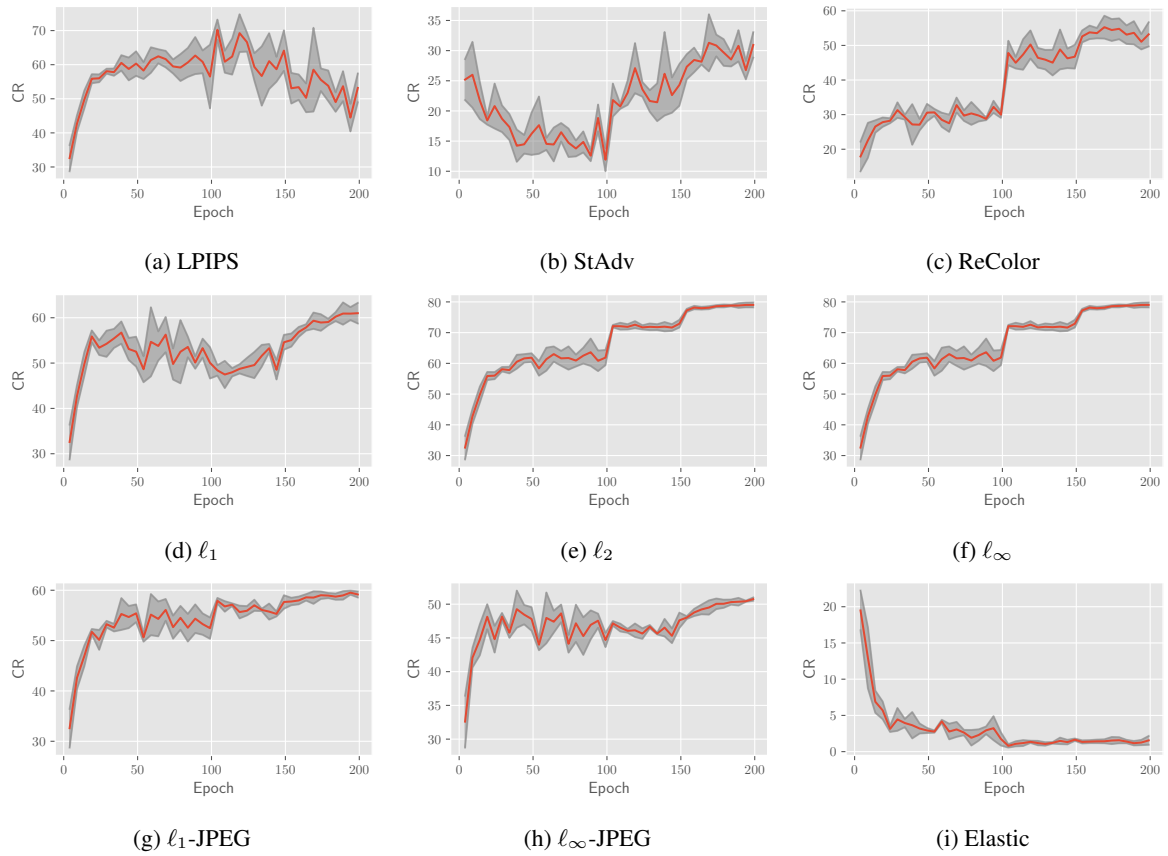
In Figure 12, we plot the impact of extra training data on $\text{CR}_{\text{ind-worst}}$ per attack. We observe that for most attacks (LPIPS, ℓ_1, ℓ_2, ℓ_∞, ℓ_1 JPEG and ℓ_∞ JPEG), extra data improves $\text{CR}_{\text{ind-worst}}$. However, for some attacks, Specifically, StAdv, ReColor, and Elastic attacks extra data does not improve $\text{CR}_{\text{ind-worst}}$. In fact, for StAdv and ReColor, the drop in performance after incorporating extra data is significant. Additionally, we find that the aggregate $\text{CR}_{\text{ind-worst}}$ trend for extra data observed in Section 5.2 is dominated by elastic attack performance.

H.3. Additional results for adversarially trained models

In this section, we present results for training with ℓ_∞, ℓ_2 threat models, and the union of ℓ_1, ℓ_2, and ℓ_∞ attacks via stochastic adversarial training (Madaan et al., 2020) analogous to those present for LPIPS threat model in Section 5.2.

H.3.1. Analysis of models trained with ℓ_∞ source threat model

Impact of architecture size In figure 13, we plot the performance of ResNet-18, ResNet-34, ResNet-50, and ResNet-101 architectures in terms of CR, clean accuracy, and stability constant. From Figures 13a and 13b, we note that while larger ResNet architectures (in particular ResNet-101) is able to achieve much higher clean accuracy, smaller architectures (ResNet-18 and ResNet-34) are able to achieve higher CR score, suggesting that smaller architectures are more optimal for multiattack robustness. Similarily, we find that these smaller architectures have smaller stability constant in Figure 13c,



(a) LPIPS (b) StAdv (c) ReColor

(d) ℓ_1 (e) ℓ_2 (f) ℓ_∞

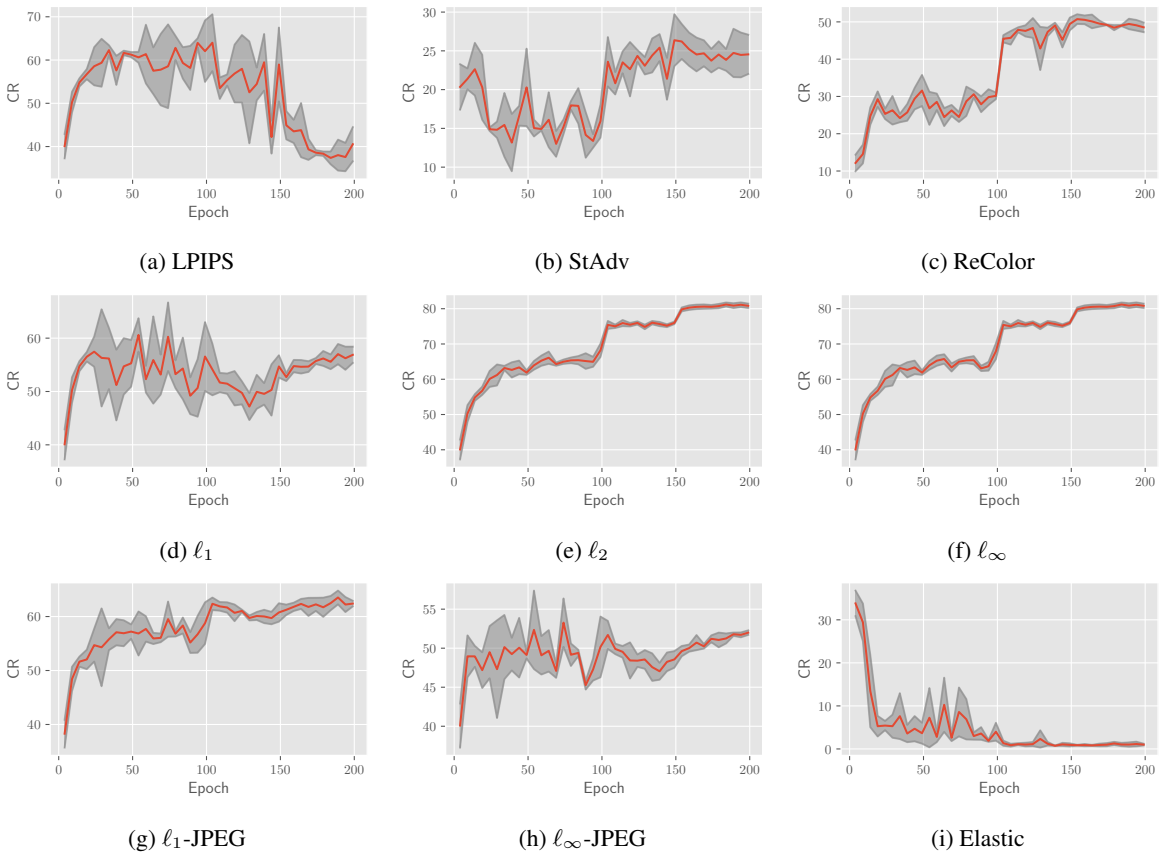
(g) ℓ_1-JPEG (h) ℓ_∞-JPEG (i) Elastic

Figure 10: Impact of number of training epochs on $\text{CR}_{\text{ind-worst}}$ per attack type for models trained using LPIPS threat model with $\epsilon = 0.5$ (via FastLPA training (Laidlaw et al., 2021))

suggesting that there is less of a performance drop when shifting to unseen attacks compared to larger architectures. This trend matches what was observed for LPIPS threat model in Section 5.2.

Impact of additional training data In Figure 14, we plot CR, clean accuracy, and stability for ResNet-18 models trained with and without additional (synthetic) training data. Similar to findings from Section 5.2, we find that additional data improves both clean accuracy and $\text{CR}_{\text{ind-avg}}$. However, there is no significant change in performance in terms of $\text{CR}_{\text{ind-worst}}$. This suggests, that while on average, extra data can improve performance across the set of tested attacks, this is not necessarily the case for worst-case performance. We find that for ℓ_∞ training, extra data does improve stability across attacks (stability constant in Figure 13c significantly decreases with extra training data), suggesting that for ℓ_∞ training, using additional data can decrease the drop in performance to unforeseen attacks.

Impact of number of epochs In Figure 15, we plot the impact of number of training epochs on CR and stability. Similar to trends for training with LPIPS threat model in Section 5.2, we find that longer training does improve average case performance. For worst-case performance, we find that $\text{CR}_{\text{ind-worst}}$ drops quickly within the first 50 epochs of training and then stays relatively constant throughout the remainder of training. This makes sense because at initialization the model is essentially randomly guessing so even on the worst-case attack, the model can still achieve about 10% robust accuracy. However, as the model trains it becomes more vulnerable to the worst-case (and likely unseen attack) leading to a large drop in worst-case robust accuracy, which causes CR to be near 0. Similar to training with LPIPS threat model, we also find that stability constant increases during training, which suggests that as training continues, the drop in robustness between seen and unseen threat models increases.



(a) LPIPS (b) StAdv (c) ReColor

(d) ℓ_1 (e) ℓ_2 (f) ℓ_∞

(g) ℓ_1-JPEG (h) ℓ_∞-JPEG (i) Elastic

Figure 11: Impact of number of training epochs on $\text{CR}_{\text{ind-worst}}$ per attack type for models trained with LPIPS threat model at $\epsilon = 0.5$ and elastic threat model at $\epsilon = 0.7$

H.3.2. Analysis of models trained with ℓ_2 source threat model

Impact of architecture size In Figure 16, we plot the CR, clean accuracy, and stability constant achieved by training ResNet-18, ResNet-34, ResNet-50, and ResNet-101. Similar to trends for training with LPIPS and training with ℓ_∞ threat model, we find that smaller architectures (ResNet-18, ResNet-34) outperform larger models in terms of CR and stability constant, suggesting that smaller models are better when it comes to multiattack robustness.

Impact of additional training data In Figure 17, we plot the clean accuracy, CR, and stability constant of ResNet-18 models trained with ℓ_2 adversarial training. Similar to what we observed for ℓ_∞ and LPIPS adversarial training, we find that extra data significantly improves $\text{CR}_{\text{ind-avg}}$, suggesting that extra data improves average robust performance over the set of attacks. However, we find that using additional data harms worst-case multiattack robustness: in Figure 17a, we find that $\text{CR}_{\text{ind-worst}}$ decreases after including additional data during training. This suggest that while on average robustness over the set of attacks increases with additional data, additional data does not uniformly improve performance over all attacks. Observing stability constant in Figure 17b, we find that extra data helps decrease tability constant, suggesting that models trained with additional data exhibit less of a drop in robustness when evaluated on attacks outside of the ℓ_2 threat model which have similar difficulty.

Impact of number of epochs In Figure 18, we plot CR and stability constant over training epochs. Similar to trends for LPIPS and ℓ_∞ threat models, we find that more training generally increases $\text{CR}_{\text{ind-avg}}$, suggesting better average case performance. Additionally, we find that $\text{CR}_{\text{ind-worst}}$ drops quickly within the first 50 epochs of training and then remains generally constant throughout the remainder of training. For stability, we find that stability constant gradually increases throughout training suggesting that as training progresses, the drop in performance across threat models increases.

H.3.3. Analysis of models trained with ℓ_1, ℓ_2, and ℓ_∞ threat models

In this section we report results for training on the union of ℓ_1, ℓ_2, and ℓ_∞ attacks via stochastic adversarial training (SAT) (Madaan et al., 2020). For these experiments, we use the same training setup as used in (Madaan et al., 2020) where for architecture size and additional data experiments we train for 30 epochs.

Impact of architecture size In Figure 19, we plot the CR, clean accuracy, and stability constant achieved by training ResNet-18, ResNet-34, ResNet-50, and ResNet-101 models. Similar to trends for training with LPIPS and training with ℓ_∞ threat model, we find that smaller architectures (ResNet-18, ResNet-34) outperform larger models in terms of CR (with ResNet-34 performing best in $\text{CR}_{\text{ind-avg}}$ and ResNet-18 performing best in $\text{CR}_{\text{ind-worst}}$) suggesting that smaller models are better when it comes to multiattack robustness when training with the union of ℓ_1, ℓ_2, and ℓ_∞ attacks. In terms of stability constant, we do not see a significant trend across architecture size.

Impact of additional training data In Figure 20, we plot the clean accuracy, CR, and stability constant of ResNet-18 models trained on the union of ℓ_1, ℓ_2, and ℓ_∞ attacks via stochastic adversarial training (Madaan et al., 2020). Similar to what we observed for other threat models used with adversarial training, we find that extra data significantly improves $\text{CR}_{\text{ind-avg}}$, suggesting that extra data improves average robust performance over the set of attacks. We also find that for this threat model, extra data also improves $\text{CR}_{\text{ind-worst}}$, which differs from the trends observed for other threat models. There does not seem to be a significant change to stability constant for this training procedure with extra data.

Impact of number of epochs In Figure 21, we plot CR and stability constant over training epochs. Similar to trends for training on other threat models, we find that more training increases $\text{CR}_{\text{ind-avg}}$, suggesting better average case performance. Additionally, we find that $\text{CR}_{\text{ind-worst}}$ drops quickly within the first 10 epochs of training and then remains gradually decreases throughout the remainder of training. For stability, we find that stability constant gradually increases throughout training suggesting that as training progresses, the change in performance across threat models increases.

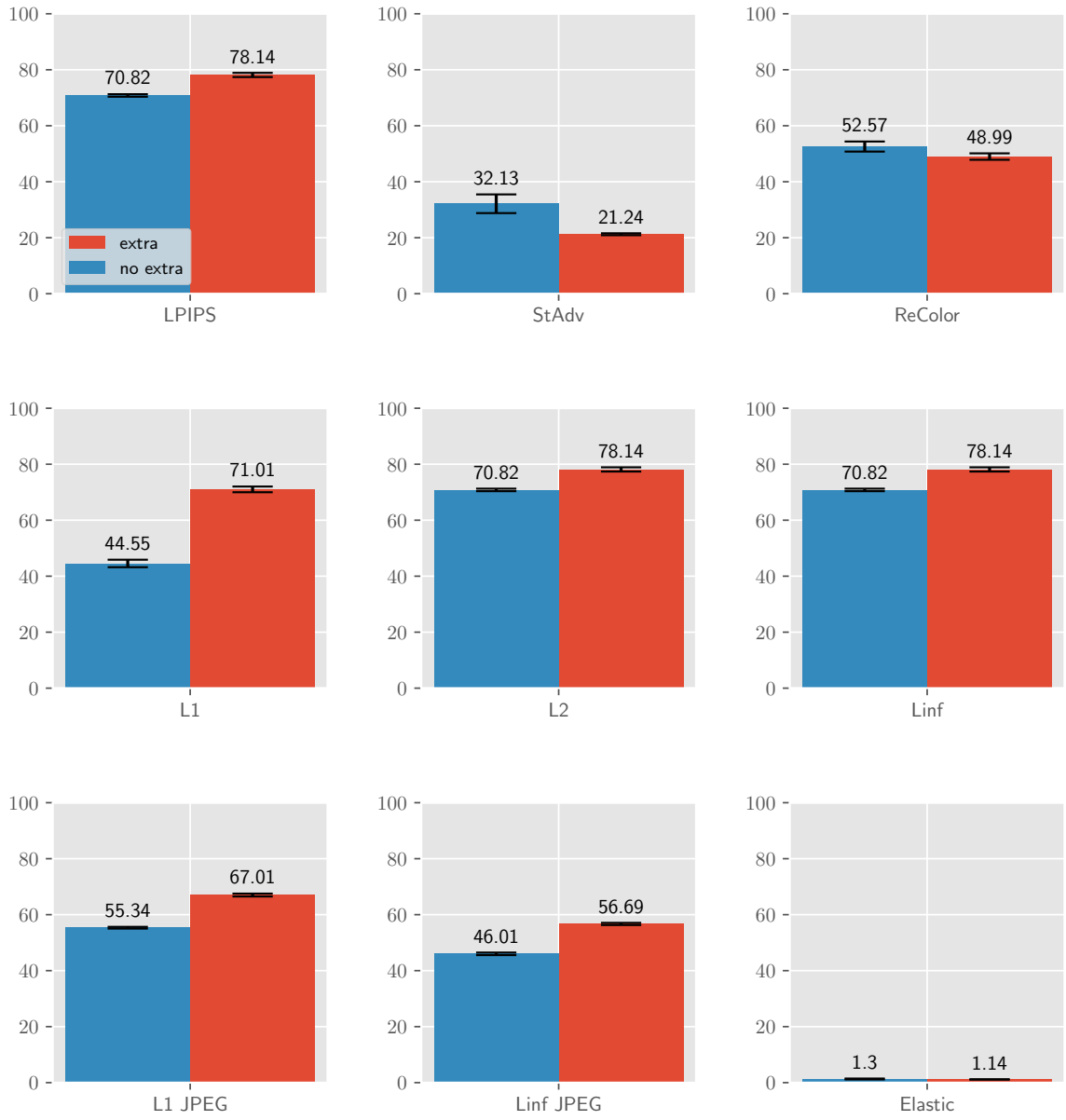
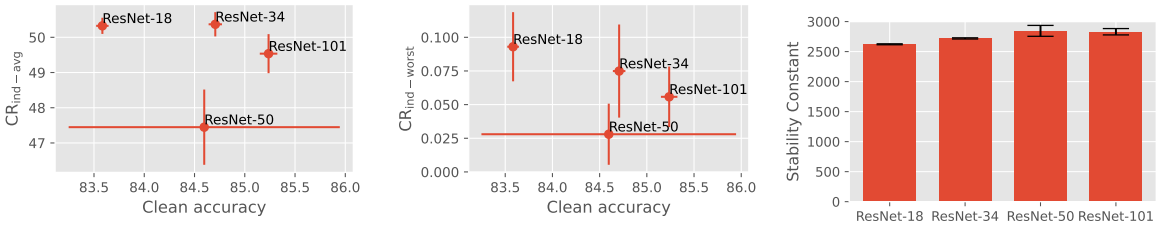
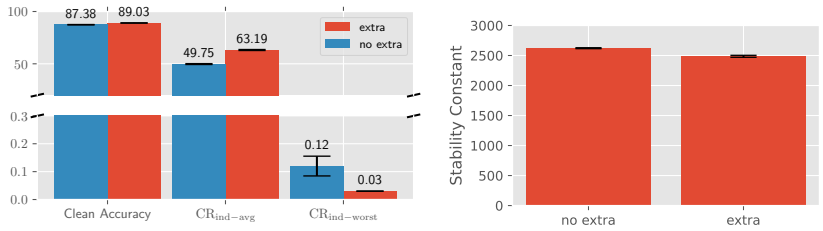


Figure 12: Impact of extra training data on $\text{CR}_{\text{ind-worst}}$ per attack type for models trained using LPIPS threat model with $\epsilon = 0.5$ (via FastLPA training (Laidlaw et al., 2021))



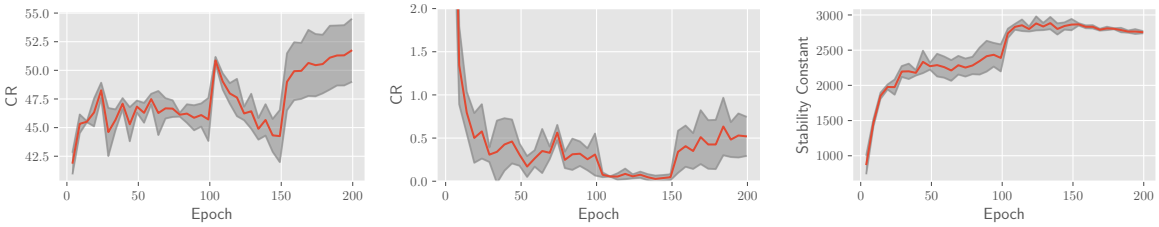
(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 13: **Impact of architecture size.** Figures (a) and (b): Clean accuracy vs CR for models trained using PGD adversarial training with ℓ_∞ threat model with radius $\frac{8}{255}$. Results are averaged over 3 trials and error bars are shown. Higher values of CR indicate better performance. Figure (c): SC computed for models of each architecture. Lower SC indicates better performance.



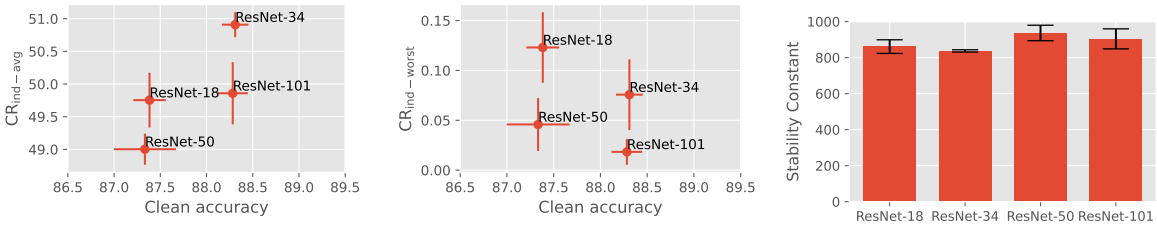
(a) Clean Accuracy and CR (b) Stability Constant

Figure 14: **Impact of additional training data.** Figure (a): Clean accuracy and CR for ResNet-18 models trained using PGD adversarial training using PGD adversarial training with ℓ_∞ threat model with radius $\frac{8}{255}$. Higher CR indicates better performance. Results are averaged over 3 trials and error bars are shown. Figure (c): SC computed for models with and without additional training data. Lower SC indicates better performance.



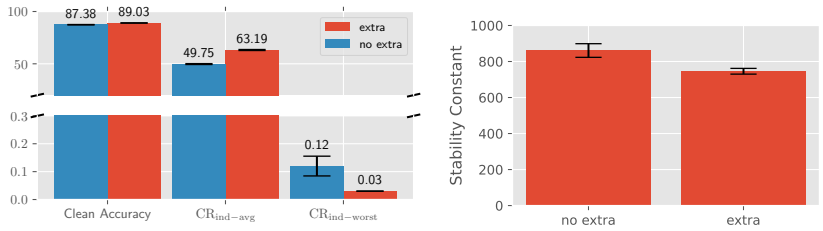
(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 15: **Impact of number of training epochs.** CR and SC over epoch for models trained with ℓ_∞ threat model with radius $\frac{8}{255}$. The red line indicates the average over 3 runs while the grey band highlights indicate 1 standard deviation from the mean. Higher values of CR and lower values of SC indicate better performance.



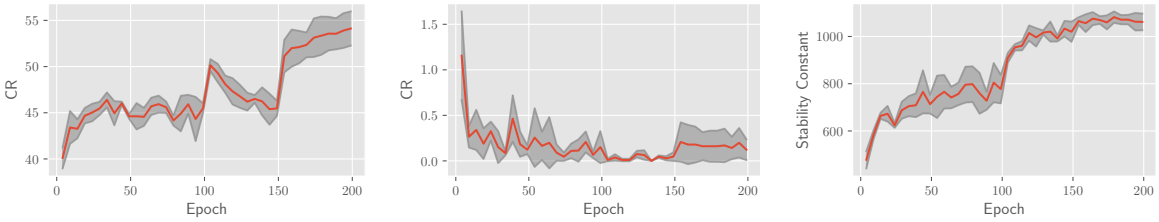
(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 16: **Impact of architecture size.** Figures (a) and (b): Clean accuracy vs CR for models trained using PGD adversarial training with ℓ_2 threat model with radius 0.5. Results are averaged over 3 trials and error bars are shown. Higher values of CR indicate better performance. Figure (c): SC computed for models of each architecture. Lower SC indicates better performance.



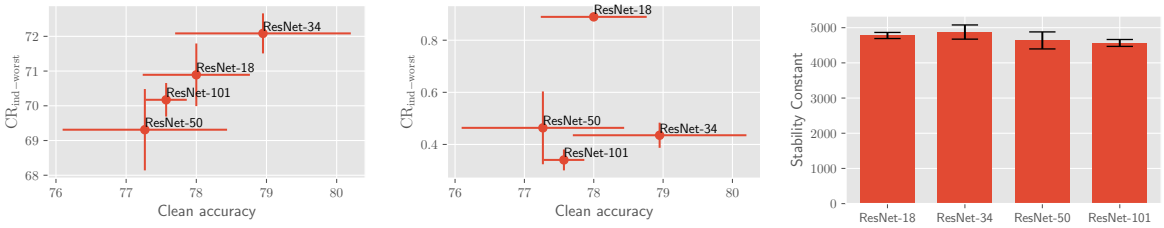
(a) Clean Accuracy and CR (b) Stability Constant

Figure 17: **Impact of additional training data.** Figure (a): Clean accuracy and CR for ResNet-18 models trained using PGD adversarial training with ℓ_2 threat model with radius 0.5. Higher CR indicates better performance. Results are averaged over 3 trials and error bars are shown. Figure (c): SC computed for models with and without additional training data. Lower SC indicates better performance.



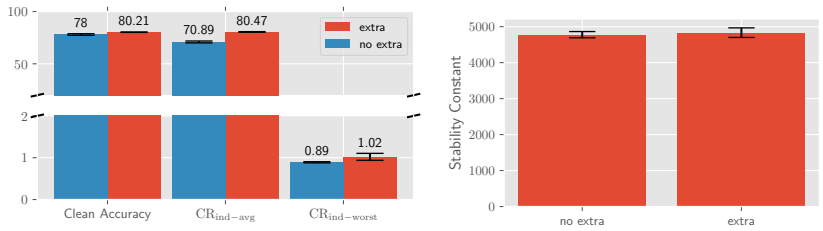
(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 18: **Impact of number of training epochs.** CR and SC over epoch for models trained with ℓ_2 threat model with radius 0.5. The red line indicates the average over 3 runs while the grey band highlights indicate 1 standard deviation from the mean. Higher values of CR and lower values of SC indicate better performance.



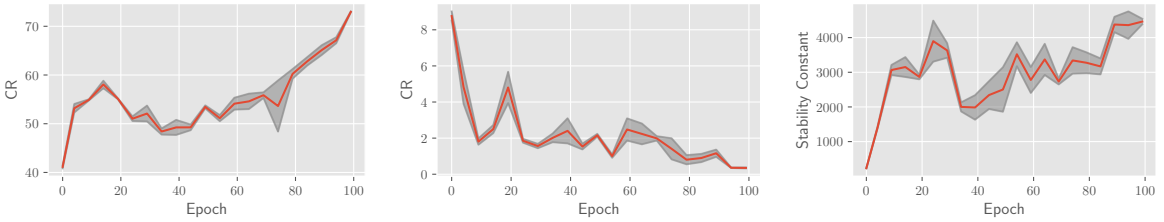
(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 19: **Impact of architecture size.** Figures (a) and (b): Clean accuracy vs CR for models trained using stochastic adversarial training (Madaan et al., 2020). Results are averaged over 3 trials and error bars are shown. Higher values of CR indicate better performance. Figure (c): SC computed for models of each architecture. Lower SC indicates better performance.



(a) Clean Accuracy and CR (b) Stability Constant

Figure 20: **Impact of additional training data.** Figure (a): Clean accuracy and CR for ResNet-18 models trained using stochastic adversarial training (Madaan et al., 2020). Higher CR indicates better performance. Results are averaged over 3 trials and error bars are shown. Figure (c): SC computed for models with and without additional training data. Lower SC indicates better performance.



(a) $CR_{ind\text{-}avg}$ (b) $CR_{ind\text{-}worst}$ (c) Stability Constant

Figure 21: **Impact of number of training epochs.** CR and SC over epoch for models trained with stochastic adversarial training (Madaan et al., 2020) The red line indicates the average over 3 runs while the grey band highlights indicate 1 standard deviation from the mean. Higher values of CR and lower values of SC indicate better performance.