Adapting to Evolving Adversaries with Regularized Continual Robust Training

Sihui Dai [*1,2] **Christian Cianfarani** [*3] **Vikash Sehwag** [4] **Prateek Mittal** [2] **Arjun Bhagoji** [5]

Abstract

Robust training methods typically defend against specific attack types, such as ℓ_p attacks with fixed budgets, and rarely account for the fact that defenders may encounter new attacks over time. A natural solution is to adapt the defended model to new adversaries as they arise via fine-tuning, a method which we call continual robust training (CRT). However, when implemented naively, fine-tuning on new attacks degrades robustness on previous attacks. This raises the question: *how can we improve the initial training and fine-tuning of the model to simultaneously achieve robustness against previous and new attacks?* We present theoretical results which show that the gap in a model's robustness against different attacks is bounded by how far each attack perturbs a sample in the model's logit space, suggesting that regularizing with respect to this logit space distance can help maintain robustness against previous attacks. Extensive experiments on 3 datasets (CIFAR-10, CIFAR-100, and ImageNette) and over 100 attack combinations demonstrate that the proposed regularization improves robust accuracy with little overhead in training time. Our findings and open-source code[1] lay the groundwork for the deployment of models robust to evolving attacks.

1. Introduction

For safety critical applications, it is important to defend machine learning (ML) models against test-time attacks. However, many existing defenses (Madry et al., 2018; Zhang et al., 2019; Croce et al., 2020) assume that the adversary is restricted to a narrow threat model such as an ℓ_p ball of fixed radius around the input. When this assumption is violated, the robustness of adversarially trained models can significantly degrade (Dai et al., 2023; Kaufmann et al., 2019). Additionally, due to rapid development of new types of attacks (Xiao et al., 2018; Laidlaw & Feizi, 2019; Laidlaw et al., 2021; Kaufmann et al., 2019), it is difficult to anticipate all types of attacks in advance. This raises the question: *how can we defend models as new attacks emerge?*

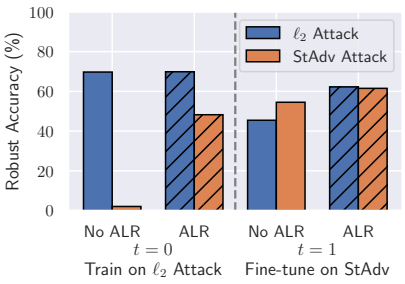


Figure 1: **Impact of our proposed regularization term (ALR) in both training and fine-tuning on CIFAR-10.** Adversarial ℓ_2 regularization (ALR) significantly improves generalization to the unforeseen StAdv attack when performing adversarial training for ℓ_2 robustness. Using ALR when subsequently fine-tuning with only StAdv attack also decreases the drop in ℓ_2 robustness.

For long-term robustness, models must quickly adapt to new attacks without sacrificing robustness to previous ones, a goal known as continual adaptive robustness (CAR) (Dai et al., 2024b) (§2). A natural approach is to apply adversarial training on known attacks and fine-tune when new ones emerge, a process we call continual robust training (CRT). However, adversarial training provides poor generalization to unseen attacks, leading to suboptimal starting points for fine-tuning, and fine-tuning itself can degrade robustness against past attacks (Figure 1).

We theoretically show that the robustness gap between attacks is linked to logit-space distances between perturbed and clean inputs and that regularizing these distances can improve generalization to new attacks and reduce drops in robustness on previous attacks. Extensive experiments confirm these findings. Our key contributions are as follows:

[*]Equal contribution [1]CapitalOne [2]Department of Electrical and Computer Engineering, Princeton University [3]Department of Computer Science, University of Chicago [4]Google Deepmind [5]Centre for Machine Intelligence and Data Science, Indian Institute of Technology, Bombay. Correspondence to: Sihui Dai <sihui.dai@capitalone.com>, Christian Cianfarani <crc@uchicago.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1]Our code is available at: https://github.com/inspire-group/continual_robust_training/

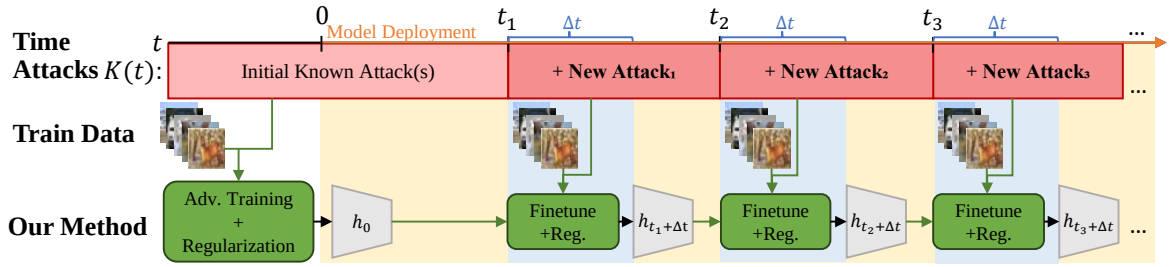


Figure 2: **An overview of the problem of adapting to new adversaries (continual adaptive robustness) and our solution framework (Regularized Continual Robust Training).** The defender learns about the existence of new attacks sequentially, and at time t aims to achieve robustness against $K(t)$, the set of attacks known at times $\leq t$. A model h_0 is deployed at time 0 to be robust against an initial set of known attacks, and new attacks are introduced at times t_1, t_2, and t_3. We propose performing regularized initial robust training on the initially known attack(s) and then using regularized fine-tuning to adapt the model against future attacks within time Δt, leading to a sequence of models $h_0, h_{t_1+\Delta t}, h_{t_3+\Delta t}, h_{t_3+\Delta t}$.

Regularized Continual Robust Training for Adapting to New Adversaries (§3) . To enhance CRT, we analyze the difference in robust losses between attacks and show it is upper bounded by the sum of the maximal ℓ_2 distance between clean and perturbed logits for both attacks. Training techniques which minimize this bound can thus improve generalization to new attacks and preserve robustness against previous ones. This motivates our proposed *adversarial ℓ_2 regularization (ALR)*, which penalizes the ℓ_2 distance between adversarial and benign logits.

Empirical Validation on Sequentially Introduced Attacks (§4.2). We conduct experiments on 2 sequences of 4 attacks across 3 datasets (CIFAR-10, CIFAR-100, and Imagenette). Our results show that ALR improves robustness in CRT with a 5.48% gain in Union accuracy (worst-case across all attacks) across ℓ_2, StAdv (Xiao et al., 2018), and Recolor attacks (Laidlaw & Feizi, 2019) over its unregularized counterpart. Figure 1 visualizes improvements brought through ALR for a sequence of 2 attacks.

Impact of ALR and Efficient Approximations in Training and Fine-Tuning (§4.3,§4.4). We conduct ablations using over 100 attack combinations (12 attack types, 9 of which are non-ℓ_p) to study ALR's role in different stages of CRT. We also explore random noise-based regularization as a more efficient alternative. We find that while noise-based regularization improves generalization in initial training, ALR is essential for maintaining robust performance during fine-tuning and improves Union accuracy by up to 7.85%.

Looking ahead (§5): We hope our methods inspire the deployment of multi-robust models against changing real-world threats. We believe our techniques could be adapted to ensure other desirable properties, such as compliance with changing standards for fairness or privacy.

2. Setup: Continual Adaptive Robustness

In this section, we introduce the problem of continual adaptive robustness (CAR) (Dai et al., 2024b), which aims to achieve robustness against new attacks as they are sequentially discovered. We survey existing approaches to this problem, with additional related work included in Appendix A. CAR is visualized in Figure 2.

2.1. A Motivating Example

Consider an entity that wants to deploy a robust ML system. The entity uses recent techniques (*e.g.* adversarial training) to defend their model against existing attack types (such as ℓ_p perturbations) and deploys their model at time $t = 0$. At a later time t_1, a research group publishes a paper about a new attack type (*e.g.* spatial perturbations (Xiao et al., 2018)) against which the entity's model is not robust. Since the ML system has been deployed, the entity would want to *quickly* modify the model to be robust against the new attack while maintaining robustness against previous attacks. Having a quick update procedure would minimize the time that an attacker can exploit this vulnerability. Quick adaptation to new attacks is the foundation of continual adaptive robustness (CAR), a problem setting introduced in a recent position paper (Dai et al., 2024b). In this work, we propose and analyze the first dedicated defense for CAR.

2.2. Problem Formulation

Notation: $\mathcal{D} = X \times Y$ denotes a data distribution where X and Y are the support of inputs and labels, respectively. $\mathcal{H}$ denotes the hypothesis class. We use $C : X \to \tilde{X}$ to define an adversarial constraint where $\tilde{X}$ is the space of adversarial examples. $\ell : Y \times Y \to \mathbb{R}$ denotes the loss function.

Attack sequences: In CAR (Dai et al., 2024b), different test-time attacks are introduced sequentially (Figure 2). Each attack P_C is associated with a constraint C and can be

formulated as a maximizer of the loss (i.e. $P_C(x, y, h) = \arg\max_{x' \in C(x)} \ell(h(x), y)$). We refer to the time at which P_C is discovered by the defender as $T(P_C)$, and the set of attacks known by the defender at a given time t as the *knowledge set* at time t: $K(t) = \{P \mid T(P) \leq t\}$. The expansion of K over time can be viewed as modeling the setting of research groups or security teams sequentially discovering new attack types.

Goals in CAR: A defender in CAR uses a defense algorithm $\mathcal{A}_{\text{CAR}}$ to deploy a model $h_t = \mathcal{A}_{\text{CAR}}(\mathcal{D}, K(t), \mathcal{H})$ at each time step t. Performance at time t is measured by Union robust loss across the knowledge set: $\mathcal{L}(h, t) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{P \in K(t)} [\ell(P(x, y, h), y)]$.

Definition 2.1 (Continual Adaptive Robustness (Dai et al., 2024b))**.** Given loss tolerances δ_{known} and δ_{unknown} with $0 < \delta_{\text{known}} < \delta_{\text{unknown}}$ and grace period Δt for recovering from a new attack, a defense algorithm $\mathcal{A}_{\text{CAR}}$ achieves CAR if for all $t > 0$:

- When $t - T(P) < \Delta t$ for any attack P and $T(P) < t$, h_t satisfies $\mathcal{L}(h_t, t) \leq \delta_{\text{unknown}}$
- Otherwise, $\mathcal{L}(h_t, t) \leq \delta_{\text{known}}$.

These criteria capture 3 distinct goals for the defender: (1) The model at time t must *achieve good robustness if no attacks have been introduced recently* (within Δt time). This is due to the δ_{known} threshold on the robust loss in the second criterion; (2) If a new attack has occurred within Δt period before the current time t, the model at time t must *achieve some robustness against the new attack*. This is modeled by the δ_{unknown} threshold in the first criterion. Since $0 < \delta_{\text{known}} < \delta_{\text{unknown}}$, CAR tolerates a degradation in robustness between the 2 cases; (3) The defense is expected to *recover robustness quickly after new attacks*. This is modeled by the Δt time window; Δt time after the introduction of a new attack, the loss threshold changes from δ_{unknown} to δ_{known}.

2.3. Baseline Approaches to CAR

CAR through multiattack robustness (MAR). Prior works for multiattack robustness often involve training with multiple attacks simultaneously (Tramèr & Boneh, 2019; Maini et al., 2020), which can be computationally expensive. A trivial (but expensive) defense algorithm for CAR is to use these training-based techniques and retrain a model from scratch on $K(t)$ every time it changes. However, this would require us to tolerate larger values of Δt.

CAR through unforeseen attack robustness (UAR). Defenses for unforeseen attack robustness (UAR) aim to ensure robustness to attacks that were not seen during training (Laidlaw et al., 2021; Dai et al., 2022). This suggests another trivial defense for CAR: use a UAR defense to get a model h and use h for all time steps. This approach is efficient since no time is spent updating the model, but would require much higher values of δ_{known} as these methods do not obtain high robustness across all attacks (Dai et al., 2023).

3. Theoretical Motivation and Methods

In this section, we introduce continual robust training (CRT) and provide theoretical results to demonstrate that adding a regularization term bounding adversarial logit distances can help balance performance across a set of adversaries.

3.1. Continual Robust Training (CRT)

Continual robust training consists of 2 parts, *initial training* and *iterative fine-tuning* (Figure 2). The output of initial training is deployed at $t = 0$ while fine-tuning is used as new attacks are introduced.

At time $t = 0$, the goal of the defender is to minimize the initial training objective: $\mathcal{L}(h, 0) = \frac{1}{m}\sum_{i=1}^{m} \ell(h(P_{C_{\text{init}}}(x_i, y_i, h)), y_i)$ where $\{(x_i, y_i)\}_{i=1}^{m}$ is the training dataset and $P_{C_{\text{init}}}$ is the initial attack. Notably, using standard training in this stage yields a high δ_{unknown}.

At $t > 0$, as new attacks are introduced, we use a fine-tuning strategy F to select the attack from $K(t)$ to use for each example. Specifically, we formulate this as: $\mathcal{L}(h, t) = \frac{1}{m}\sum_{i=1}^{m} \ell(h(P_C(x_i, y_i, h)), y_i)$ where $P_C = F(K(t), (x_i, y_i))$. Fine-tuning strategies include picking the attack that maximizes $\ell(x_i, y_i)$, randomly sampling from $K(t)$, and using the newest attack. A good fine-tuning strategy would be able to quickly adapt the model to new attacks, allowing it to satisfy a small Δt threshold. However, naive fine-tuning does not guarantee good performance across all attacks and may require large values of δ_{known}. As illustrated in Figure 1, a model may lose robustness to the initial attack after the fine-tuning stage. We now discuss how such degradation can be addressed through regularization.

3.2. Bounding the Difference in Adversarial Losses

A successful implementation of CAR would both enhance the robustness of a model to new attacks encountered at a given time step and maintain robustness to attacks seen at previous time steps. We will show how the gap in robustness between attacks relates to distances between adversarially perturbed representations in the logit space of a model, which suggests the use of regularization as a tool for bounding the impact of any given attack.

Let $h : \mathbb{R}^d \to \mathbb{R}^k$ be a k class neural network classification model. To simplify the problem setting, we focus on the state of the model when attacks P_{C_1} and P_{C_2} (with corresponding adversarial constraints C_1 and C_2) are known to the defender. Consider the following two adversarial loss functions: $\mathcal{L}_1(h) := \mathbb{E}_{\mathcal{D}}[\ell(h(P_{C_1}(x, y)), y)]$ and $\mathcal{L}_2(h) := \mathbb{E}_{\mathcal{D}}[\ell(h(P_{C_2}(x, y)), y)]$. Without loss of gener-

ality, assume that $\mathcal{L}_1(h) \geq \mathcal{L}_2(h)$. We can then bound the difference between $\mathcal{L}_1(h)$ and $\mathcal{L}_2(h)$, adapting a result from Nern et al. (2023), as follows[2]:

Theorem 3.1. *Assume that loss $\ell(\hat{y}, y)$ is M_1-Lipschitz in $\|\cdot\|_2$, for $\hat{y} \in h(X)$ with $M_1 > 0$ and bounded by $M_2 > 0$ [3], i.e. $0 \leq \ell(\hat{y}, y) \leq M_2 \; \forall \hat{y} \in h(X)$. Then, for a subset $\mathbb{X} = \{x_i\}_{i=1}^n$ independently drawn from $\mathcal{D}$, the following holds with probability at least $1 - \rho$:*

$$\mathcal{L}_1(h) - \mathcal{L}_2(h) \leq M_1 \frac{1}{n} \sum_{i=1}^{n} \Bigg(\max_{x' \in C_1(x_i)} \|h(x') - h(x_i)\|_2 + \max_{x' \in C_2(x_i)} \|h(x') - h(x_i)\|_2 \Bigg) + D,$$

where $D = M_2 \sqrt{\frac{\log(\rho/2)}{-2n}}$.

This result suggests that regularization with respect to a single attack (say, in pre-training) will give the model greater resiliency against unforeseen attacks and help meet the δ_{unknown} threshold. Using regularization when fine-tuning on a new attack could also prevent degradations in robustness against previously seen attacks, helping to meet the δ_{known} threshold. Using similar reasoning, we can also bound the gap between Union and clean loss:

Corollary 3.2. *Let $\mathcal{L}_{1,2}(h) := \mathbb{E}_{\mathcal{D}} \left[\max \left(\ell(h(P_{C_1}(x, y, h)), y), \ell(h(P_{C_2}(x, y, h)), y) \right) \right]$. Then, with probability at least $1 - \rho$,*

$$\mathcal{L}_{1,2}(h) - \mathcal{L}(h) \leq M_1 \frac{1}{n} \sum_{i=1}^{n} \Bigg(\max_{x' \in C_1(x_i)} \|h(x') - h(x_i)\|_2 + \max_{x' \in C_2(x_i)} \|h(x') - h(x_i)\|_2 \Bigg) + D.$$

This corollary helps characterize the trade-off between clean and robust loss in our setting. Although our results are stated in terms of pairs of attacks, Theorem 3.1 and Corollary 3.2 straightforwardly lead to meaningful bounds for larger sets of attacks. Theorem 3.1 upper bounds the maximum gap in robust loss between any pair of attacks in the set, and Corollary 3.2 upper bounds the gap between the clean loss and the Union loss on all attacks. Proofs of Theorem 3.1 and Corollary 3.2 are present in Appendix D.

Comparison to Dai et al. (2022): We note that Dai et al. (2022, Theorem 4.2) derive a related bound on the adversarial loss gap between two attacks in the context of UAR. However, their formulation assumes that the constraint set of the target attack is a strict superset of that of the source attack, whereas we make no assumptions about the relationship between the two constraint sets.

[2] As stated, these results hold for loss functions that are Lipschitz with respect to the ℓ_2 norm. We note that similar bounds can be derived for other norms by applying a constant scaling factor to the first term of the bound (i.e. for losses Lipschitz with respect to the ℓ_1 norm, the scaling factor would be $\sqrt{c}$).

[3] We note that surrogate losses such as the cross-entropy used during training are not bounded, but the $0-1$ loss which is often the key quantity of interest *is bounded*.

3.3. Regularization Methods

Theorem 3.1 suggests that reducing the sensitivity of logits to *either* attack has the potential to reduce the performance gap between attacks (see Figure 4 in the Appendix for an empirical validation of this effect). To this end, we propose incorporating regularization into both training stages. Specifically, we adopt modified training objective $\mathcal{L}_{\text{reg}}(h, t) = \mathcal{L}(h, t) + \lambda R(h, K(t))$, where λ is the regularization strength and $R(h)$ is the regularization term used. We will now discuss several forms of regularization.

Adversarial ℓ_2 regularization. (ALR) Driven by our theoretical results, we first introduce adversarial ℓ_2 regularization: $R_{\text{ALR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \max_{x' \in C(x_i)} \|h(x') - h(x_i)\|_2$ where $C = C_{\text{init}}$ in initial training and corresponds to attack $P_C = F(K(t), (x_i, y_i))$ chosen by the fine-tuning strategy. ℓ_2 regularization penalizes the maximum distance between a sample's logits and the furthest adversarially perturbed logits within that sample's neighborhood. Using this regularization term would directly minimize the upper bounds in Theorem 3.1 and Corollary 3.2. We note that while ALR is similar in form to TRADES (Zhang et al., 2019), it uses a Euclidean distance instead of the KL-divergence. Our paper is the first to show that this form of regularization is beneficial for CAR.

Efficiently approximating ALR. Computing ALR uses multi-step optimization which can be costly to compute in practice. To improve efficiency in experiments, we consider (1) using single step optimization for ALR and (2) using randomly sampled, unoptimized perturbations can help with CAR. For (2), we consider Gaussian noise regularization (GR) and Uniform noise regularization (UR), specifically: $R_{\text{GR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \|h(x') - h(x_i)\|_2$ where $x' \sim \mathcal{N}(0, \sigma^2)$ and $R_{\text{UR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \|h(x') - h(x_i)\|_2$ where $x' \sim \mathcal{U}(-\sigma, \sigma)$.

Other Regularizers. We compare to variation regularization (VR), which has been shown to improve generalization to unforeseen attacks (Dai et al., 2022). VR is defined as: $R_{\text{VR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \max_{x', x'' \in C(x_i)} \|h(x') - h(x'')\|_2$ where $C = C_{\text{init}}$ in initial training. We also consider VR in finetuning with C corresponding to attack $P_C = F(K(t), (x_i, y_i))$. The link between VR and ALR is discussed in Appendix E.

We compare to the TRADES regularizer (Zhang et al., 2019) during *initial training* of the model. This regularizer can be formulated as $R_{\text{TRADES}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \max_{x' \in C(x_i)} KL(h(x'), h(x_i))$ and measures the

worst case KL-distance between the logit distributions after a perturbation is applied.

For *fine-tuning*, we consider elastic weight consolidation (EWC) (Kirkpatrick et al., 2017), a technique for reducing catastrophic forgetting in continual learning. EWC ensures that the model parameters do not deviate too much from the previous task (or in our case, attack) learned. Mathematically, $R_{\text{EWC}}(h, K(t)) = \sum_i \frac{1}{2} F_i(\theta_i - \theta^*_{\text{prev},i})^2$ where F is the diagonal of the Fisher information matrix, θ is the model parameters that we are optimizing, θ^*_{prev} are the parameters of the model that we are fine-tuning from.

4. Experimental Results

In this section, we empirically demonstrate that using regularization in CRT helps improve robustness when attacks are introduced sequentially. This section is organized as follows: (i) experimental setup §(4.1), (ii) overall results for using regularization in CRT (§4.2), (iii) ablations in initial training (§4.3) and (iv) ablations in fine-tuning (§4.4).

4.1. Experimental Setup

Datasets. We experiment with CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ImageNette (Howard), a 10-class subset of ImageNet (Deng et al., 2009).

Architectures. For CIFAR-10 and CIFAR-100, we use WideResnet-28-10 (WRN-28-10) architecture (Zagoruyko & Komodakis, 2016) and ResNet-18 for ImageNette.

Attacks. We include results for ℓ_2, ℓ_∞, StAdv (Xiao et al., 2018), ReColor attacks (Laidlaw & Feizi, 2019), and the 8 core attacks of Imagenet-UA (Kaufmann et al., 2019). For ℓ_2 attacks, we use a bound $\epsilon = 0.5$ for CIFAR datasets and $\epsilon = 1$ for ImageNette. For ℓ_∞ attacks, we use $\epsilon = \frac{8}{255}$, and for StAdv and ReColor attacks, we use the same bounds as used in their original papers Xiao et al. (2018) ($\epsilon = 0.05$) and Laidlaw & Feizi (2019) ($\epsilon = 0.06$) respectively. For ImageNet-UA attacks, we use the *medium distortion* strength bounds used by Kaufmann et al. (2019). For experiments investigating the impact of regularization in the fine-tuning step of CRT (§4.4), we include results for fine-tuning to the same attack type but with larger attack bounds. For these experiments, the larger bounds are given by $\epsilon = 1$ for ℓ_2, $\epsilon = \frac{12}{255}$ for ℓ_∞, $\epsilon = 0.07$ for StAdv, $\epsilon = 0.08$ for ReColor, and high distortion strength bounds for ImageNet-UA attacks.

Training from scratch baselines. We consider the following baselines for training from scratch:

- *Training with AVG and MAX objectives* (Tramèr & Boneh, 2019): Tramèr & Boneh (2019) propose two different training objectives, AVG ($L_{\text{AVG}}(h,t) = \frac{1}{m|K(t)|} \sum_{i=1}^m \sum_{P_C \in K(t)} \ell(h(P_C(x_i, y_i)), y_i)$) and MAX ($L_{\text{MAX}}(h,t) = \frac{1}{m} \sum_{i=1}^m \max_{P_C \in K(t)} \ell(h(P_C(x_i, y_i)), y_i)$), for robustness against multiple known attacks.
- *Randomly sampling attacks* (Madaan et al., 2020): AVG and MAX require generating adversarial examples with all attacks for each image. For a more efficient baseline, we consider randomly sampling an attack for each batch for use in adversarial training.

CRT Baselines. For CRT, we use PGD adversarial training (AT) (Madry et al., 2018) for initial training and then fine-tune the model using several different fine-tuning strategies:

- *MAX objective fine-tuning* (FT-MAX) (Tramèr & Boneh, 2019): We use the MAX objective for fine-tuning when a new attack is introduced.
- *Croce & Hein (2022) fine-tuning* (FT Croce): Croce & Hein (2022) introduce a fine-tuning technique for use with ℓ_∞ and ℓ_1 attacks which we *generalize to training with arbitrary attacks*. This approach samples a single attack per batch. The probability that an attack P_C is sampled is given by $\frac{\text{err}(P_C)}{\sum_{P \in K(t)} \text{err}(P)}$ where $\text{err}(P)$ denotes the running average of robust loss with respect to attack P computed across batches of each attack.
- *Single attack fine-tuning* (FT Single): We also consider fine-tuning with *only the newly introduced attack*, allowing us to determine the extent to which previous attacks are forgotten. The previous two fine-tuning techniques involve replaying previous attacks.

We then investigate incorporating regularization into the initial training and fine-tuning phases of CRT.

Training and Fine-tuning Procedures. During training, we use 10-step Projected Gradient Descent (Madry et al., 2018) to generate adversarial examples. For the regularization terms (§3.3), VR and ALR use single step optimization to reduce time overhead, while UR and GR use $\sigma = 2$ and $\sigma = 0.2$, respectively. Results for additional values of σ are in Appendix I.1. We train models for 100 epochs for initial training and 10 epochs for fine-tuning (results with 25 epochs in Appendix H). We include additional details about the training procedure in Appendix G.

Evaluation Attacks and Metrics. Our main results in Table 1 and additional ones in Appendix H use full AutoAttack (Croce & Hein, 2020b) for evaluating ℓ_p robustness. For ablations, we restrict to APGD-T and FAB-T from AutoAttack to reduce evaluation time. We use 20-step optimization when evaluating StAdv and ReColor attacks and the default evaluation hyperparameters for ImageNet-UA attacks in Kaufmann et al. (2019). We report *accuracy on each attack*, *Union accuracy* (overall accuracy when the worst case attack is chosen for each test example), *Average accuracy* (average over accuracy on each attack), and *training time* (in hours). Metrics are reported for the epoch

Time Step		Procedure	Threat Models	Clean	ℓ_2	StAdv	ℓ_∞	Recolor	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	Init	AT	ℓ_2	**91.17**	69.7	2.08	28.41	44.94	69.7	69.7	36.28	1.24	8.68
		AT + ALR ($\lambda = 1$)	ℓ_2	89.43	**69.84**	**48.23**	**34.00**	**65.46**	**69.84**	**69.84**	**54.38**	**31.27**	17.17
1	Finetune	FT MAX	ℓ_2, StAdv	83.73	57.07	58.67	12.51	49.03	57.87	51.32	44.32	12.36	4.00
		FT Single	ℓ_2, StAdv	80.89	45.45	54.5	6.09	41.98	49.98	41.05	37.0	5.87	2.78
		FT Croce	ℓ_2, StAdv	84.7	57.88	54.27	14.38	51.08	56.07	48.13	44.4	13.8	2.40
		FT Single + ALR	ℓ_2, StAdv	**87.24**	**62.22**	61.5	**21.4**	**70.87**	61.86	55.04	**54.0**	**21.14**	4.24
		FT Croce + ALR	ℓ_2, StAdv	86.03	59.18	**65.14**	15.36	63.31	**62.16**	**55.83**	50.75	15.29	3.47
2	Finetune	FT MAX	ℓ_2, StAdv, ℓ_∞	83.16	65.63	56.68	36.9	65.69	53.07	35.18	56.23	34.83	5.62
		FT Single	ℓ_2, StAdv, ℓ_∞	87.99	**70.53**	11.17	41.63	63.46	41.11	7.95	46.7	7.74	1.57
		FT Croce	ℓ_2, StAdv, ℓ_∞	85.05	67.3	48.07	33.38	62.52	49.58	28.96	52.82	28.63	2.27
		FT Single + ALR	ℓ_2, StAdv, ℓ_∞	**88.74**	69.15	47.33	**42.08**	68.62	52.85	**36.66**	56.8	**36.62**	2.26
		FT Croce + ALR	ℓ_2, StAdv, ℓ_∞	86.57	67.99	**61.55**	36.59	**72.16**	**55.38**	35.68	**59.57**	35.52	2.87
3	Finetune	FT MAX	ℓ_2, StAdv, ℓ_∞, Recolor	83.64	66.21	57.53	**37.77**	69.32	57.71	**36.02**	57.71	**36.02**	8.45
		FT Single	ℓ_2, StAdv, ℓ_∞, Recolor	90.41	66.47	3.93	29.6	69.03	42.26	2.49	42.26	2.49	3.11
		FT Croce	ℓ_2, StAdv, ℓ_∞, Recolor	86.64	**68.76**	44.81	36.02	68.05	54.41	29.44	54.41	29.44	2.34
		FT Single + ALR	ℓ_2, StAdv, ℓ_∞, Recolor	**90.45**	61.58	25.77	27.43	69.26	46.01	19.2	46.01	19.2	4.24
		FT Croce + ALR	ℓ_2, StAdv, ℓ_∞, Recolor	87.62	68.14	**58.5**	36.39	**72.35**	**58.85**	34.92	**58.85**	34.92	3.35

Table 1: **Continual Robust Training on CIFAR-10.** Best performance for each time step are **bolded**. The defender initially knows about ℓ_2 attacks and over time, is sequentially introduced to StAdv, ℓ_∞, and ReColor attacks. We report clean accuracy, accuracy on individual attacks, and average and union accuracies. The "Threat Models" column specifies known attacks at the current time step, and accuracies on these attacks are in green cells. Initial adversarial training occurs at time step 0, and the model is updated through fine-tuning the model from the previous time step. "Avg (known)" and "Union (known)" columns represent average and union accuracies on known attacks while "Avg (all)" and "Union (all)" columns report performance across all four attacks. We report training time for each time step in the "Time" column.

E^* with best performance on the set of known attacks. For training from scratch, the reported training time is scaled by fraction of training for the best epoch (*i.e.* we report $\frac{E^*}{100} \times$ training time for 100 epochs). For fine-tuning we report training time for the full 10 epochs. This allows us to see how much faster fine-tuning is to optimal early stopping when re-training from scratch.

Procedure	Clean	Avg	Union	Time
MAX	84.3	54.18	37.44	61.09
AVG	87.77	54.5	30.39	51.55
Random	86.32	54.27	30.76	13.15
CRT + ALR	87.62	58.85	34.92	26.86

Table 2: Regularized CRT (using Croce & Hein (2020b) fine-tuning strategy) compared to training from scratch on ℓ_2, StAdv, ℓ_∞, and Recolor attacks on CIFAR-10.

4.2. Improving CRT with Regularization

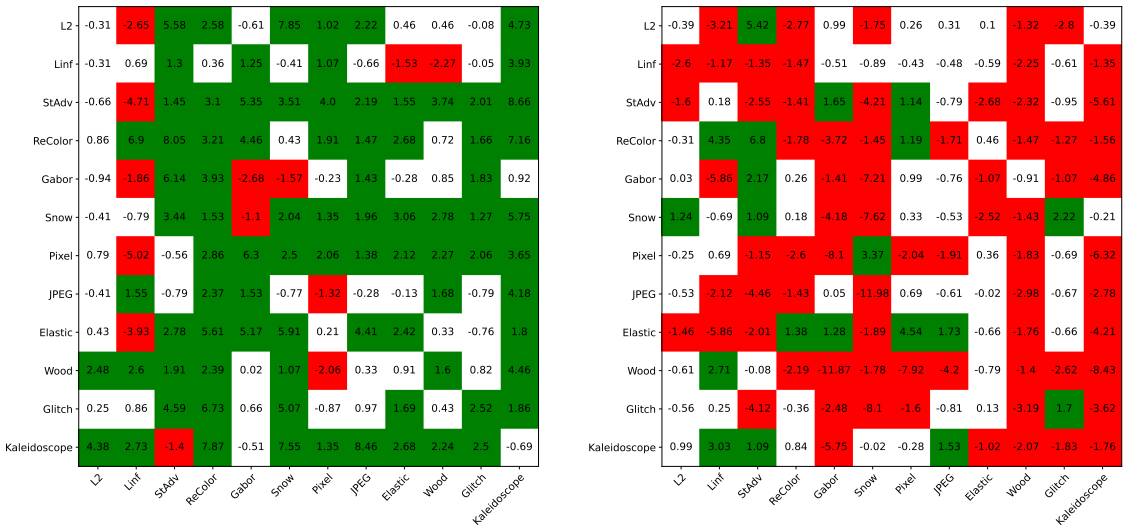
We now analyze the robustness of models trained using CRT with and without regularization. For simplicity, we focus on ALR with other methods analyzed in §4.3. To model a CAR setting, we consider a sequence of 4 attacks: $\ell_2 \rightarrow$ StAdv $\rightarrow \ell_\infty \rightarrow$ Recolor. The first attack is the initially known attack while other attacks are introduced at later time steps. We present results for CIFAR-10 in Table 1. We include results in Appendix H for Imagenette and CIFAR-100 as well as additional results for longer duration of fine-tuning (25 epochs) and a separate sequence of attacks: $\ell_\infty \rightarrow$ StAdv $\rightarrow$ Recolor $\rightarrow \ell_2$. For these experiments, we use $\lambda = 0.5$ unless specified otherwise.

Regularization reduces degradation on previous attacks. From Table 1, we observe that fine-tuning with only the new attack (FT Single) can lead to degradation of robustness against previous attacks. The incorporation of ALR significantly decreases this drop in robustness. For example, when fine-tuning from an ℓ_2 robust model with StAdv attacks (time step 1 in Table 1), FT Single incurs a 24.25% drop (from 69.7% to 45.45%) in ℓ_2 accuracy from the initial checkpoint (AT at time step 0). Meanwhile FT Single + ALR only experiences a 7.62% drop (from 69.84% to 62.22%) in ℓ_2 accuracy from the initial checkpoint (AT + ALR at time step 0). Similarly, after the introduction of ℓ_∞ attack at time step 2, the accuracy of FT Single on StAdv attacks drops 43.42% (from 54.5% to 11.17%) while FT Single + ALR only experiences a 14.17% drop (from 61.5% to 47.33%). These results align with Theorem 3.1: when incorporating ALR into training, the gap in loss on the two attacks is lessened.

Regularization improves performance on held out (unforeseen) attacks. We observe that regularized CRT leads to higher robustness on attacks held out from training. For example, at time step 1 in Table 1, which trains with ℓ_2 and StAdv attacks, the best accuracy on Recolor attacks out of unregularized CRT methods is 51.08%, while FT Single + ALR achieves 70.87% accuracy on Recolor attacks and FT Croce + ALR achieves 63.31% accuracy on Recolor attacks. The improvement in robustness on unforeseen attacks aligns with Corollary 3.2 as regularization helps decrease the drop in accuracy between clean inputs and perturbed inputs. This also aligns with CAR's goal of having a small δ_{unknown}.

Initial Attack	Reg Type	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	None	0	91.08	70.02	29.38	0.79	33.69	66.93	24.59	14.99	64.22	45.13	70.85	80.30	30.08	44.25	0.10
ℓ_2	VR	0.2	89.99	70.38	34.56	13.41	48.99	67.64	29.09	22.57	66.64	48.38	73.31	80.07	32.33	48.94	5.40
ℓ_2	ALR	0.5	89.57	70.29	34.16	17.44	51.04	65.63	28.71	22.50	66.76	48.80	73.24	79.66	28.83	48.92	5.94
ℓ_2	UR	5	88.34	66.66	27.41	26.22	60.22	69.16	26.67	22.57	64.08	46.83	71.14	77.60	31.36	49.16	6.23
ℓ_2	GR	0.5	86.89	68.19	32.02	16.54	58.32	74.85	25.69	21.26	65.32	46.82	74.08	76.99	31.93	49.33	4.18
ℓ_∞	None	0	85.53	59.36	50.98	6.34	56.27	68.94	36.79	20.57	54.02	51.00	64.24	75.94	39.44	48.66	1.31
ℓ_∞	VR	0.2	82.58	58.36	51.53	18.98	62.12	67.18	39.22	23.62	54.73	52	63.35	71.72	43.18	50.50	5.08
ℓ_∞	ALR	0.5	83.18	58.21	51.47	19.50	61.02	68.75	37.94	22.78	53.89	49.82	63.47	73.57	39.88	50.02	5.52
ℓ_∞	UR	5	78.04	60.28	40.59	42.25	70.00	67.06	33.40	26.57	60.07	49.21	64.61	67.08	38.43	51.63	8.36
ℓ_∞	GR	0.5	80.65	59.74	46.12	34.57	70.49	68.33	35.80	26.04	57.28	51.98	65.46	70.73	38.21	52.06	6.28

Table 3: **Impact of Regularization on Unforeseen Robustness.** We consider the setting where the defender is only aware of a single attack and performs training with and without different types of regularization: variation regularization (VR), adversarial ℓ_2 regularization (ALR), uniform regularization (UR), and Gaussian regularization (GR) at regularization strength λ. We report clean accuracy and robust accuracies on a range of attacks. Green cells represent an improvement of at least 1% while red cells represent a drop of at least 1% in comparison to no regularization.



(a) Adversarial ℓ_2 regularization ($\lambda = 0.5$)

(b) Uniform Regularization ($\sigma = 2, \lambda = 1$)

Figure 3: **Ablation 2: Change in union robust accuracy after fine-tuning with regularization (initial model does not use regularization).** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model *fine-tuned with and without regularization*. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% in red. Further results are in Appendix J.

Regularization balances performance and efficiency. Our proposed regularization term adds a small computational overhead over other FT approaches but generally improves union performance on the set of known attacks. For example, when considering the sequence of ℓ_2 and StAdv attacks (time step 1 in Table 1), FT Croce + ALR improves union accuracy over FT Croce by 7.7% while adding a time overhead of 1.07 hours. Additionally, when considering the sequence of 3 attacks (ℓ_2, StAdv, and ℓ_∞ attacks), FT Croce + ALR improves union accuracy over FT Croce by 6.72% while adding a time overhead of 0.6 hours. This increase in time complexity is much smaller than FT MAX which takes 1.6 hours longer than FT Croce for ℓ_2 and StAdv and 3.35 hours longer for ℓ_2, StAdv, and ℓ_∞. With respect to goals in CAR, regularization balances δ_{known} and Δt.

Comparison to training from scratch. In Table 2, we report clean, average, and union accuracies along with total training times for using training from scratch on all 4 attacks compared to training sequentially with regularized CRT on CIFAR-10. We observe that regularized CRT is significantly more efficient than MAX and AVG training (taking a total of 26.86 hours while AVG and MAX take over 50 hours of training time). Surprisingly, we find that on CIFAR-10, regularized CRT can outperform training from scratch methods, achieving 4.35% higher average accuracy compared to the best achieved by training from scratch. This suggests that transferable robustness between carefully chosen attacks can improve MAR as a whole. However, we note that the ability to outperform training from scratch seems to be specific to CIFAR-10; for ImageNette and CIFAR-100 (Appendix H)

training from scratch outperforms using fine-tuning in CAR.

Impact of dataset and attack sequence. In Appendix H, we provide results on ImageNette and CIFAR-100 as well as for attack sequence $\ell_\infty \to$ StAdv $\to$ Recolor $\to \ell_2$. Overall, we observe that trends such as improved robustness to unforeseen and the union of attacks are generally consistent. However, but the extent to which regularization improves performance over FT Croce varies. The choice of the initial attack seems to play a role in subsequent robustness, and if defenders are aware of multiple attacks, choosing the right one to start with is an interesting open question.

TAKEAWAY 1. CRT+ALR improves robustness on both known and unforeseen attacks, and reduces drop in robustness on previous attacks with only a small overhead in fine-tuning time compared to unregularized CRT.

4.3. Ablation 1: Regularization in Initial Training

We now study the impact of regularization *only* in the initial training phase of CRT. In Table 3, we present results for robust accuracies of models initially trained on ℓ_2 and ℓ_∞ attacks with different forms of regularization. We present results for different regularization strengths and initial attack choices in Appendix I.3.

Regularization improves robustness on unforeseen attacks. Interestingly, we find that all regularization types including random noise-based regularization can improve unforeseen robustness. For example, at $\lambda = 5$, UR improves union accuracy across all attacks by 6.13% for ℓ_2 initial attack and by 7.05% for ℓ_∞ initial attack compared to the model trained without regularization. Improved unforeseen robustness provides a better starting point for fine-tuning, which we demonstrate experimentally in Appendix I.4.

Trade-offs for clean and different attack accuracies. We observe that all regularization types generally exhibit a trade-off with clean accuracy and trade-offs with a few attack types such as Glitch. This trade-off aligns with Corollary 3.2 which states that the gap between clean loss and loss over the union of attacks is decreased via regularization. We also find that random noise based regularization (UR and GR) generally exhibits trade-off with the robust accuracy on the initial attack. This is generally not the case for adversarial regularization (ALR and VR) which maintains performance on the initial attack.

Regularized initial models are better starting points for fine-tuning. In Appendix I.4, we present results for fine-tuning with a new attack from models using regularization in only initial training. We observe that for all regularization types, regularization in initial training can improve the robustness on the union of attacks after fine-tuning, but this trend is more consistent with adversarial regularization types (ALR and VR) compared to random regularization types (UR and GR).

Comparison to TRADES. In Table 4, we compare ALR at $\lambda = 1$ for ℓ_2 and $\lambda = 0.5$ for ℓ_∞ to TRADES regularizer at $\lambda = 6$. Results for other strengths of TRADES regularizer in Appendix I. We observe that TRADES regularizer can also help improve unforeseen robustness but ALR is generally more effective. We also find that for ℓ_∞ initial attack, TRADES heavily trades off robustness on ℓ_∞ and ℓ_2 attacks in order to obtain higher ReColor attack accuracy.

Initial Attack	Reg Type	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Union
ℓ_2	None	**91.17**	69.7	28.41	2.08	44.94	1.24
ℓ_2	TRADES	88.76	69.69	33.00	7.04	56.82	5.51
ℓ_2	ALR	89.43	**69.84**	**34.00**	**48.23**	**65.46**	**31.27**
ℓ_∞	None	**85.93**	**59.48**	51.44	14.87	62.48	11.9
ℓ_∞	TRADES	85.72	56.44	41.70	23.17	**70.23**	17.83
ℓ_∞	ALR	83.18	58.15	**51.49**	**34.78**	58.15	**29.87**

Table 4: **Comparison to TRADES.** We compare robustness measured across different threat models when initial training on ℓ_2 and ℓ_∞ with either TRADES at $\lambda = 6$ and or ALR at $\lambda = 1$ for ℓ_2 and $\lambda = 0.5$ for ℓ_∞ regularizer.

TAKEAWAY 2. Adversarial and random noise regularization in initial training improves performance on unforeseen attacks. Fine-tuning on a new attack from a regularized model boosts resulting Union accuracy.

4.4. Ablation 2: Regularization during Fine-tuning

We now investigate whether regularization within just the the fine-tuning phase can improve CAR. We initially train models on a single initial attack using adversarial training (*without regularization*) and then fine-tune with Croce & Hein (2022)'s fine-tuning approach both with and without regularization on a new attack. In Figure 3, we present grids representing differences in Union accuracy between regularized and unregularized fine-tuning. Rows represent the initial attack used to adversarially train the model (without regularization), columns represent the new attack. We provide corresponding plots detailing differences in average accuracy, initial attack accuracy, new attack accuracy, and clean accuracy in Appendix J.1.

Adversarial regularization can improve union accuracy in fine-tuning. We find that across different initial and new attack pairs, using ALR in fine-tuning generally improves union accuracy as most cells in Figure 3(a) are green. These increases in robustness can be quite large; for example, when the initial attack is StAdv (Xiao et al., 2018) and the new attack is Kaleidoscope (Kaufmann et al., 2019), ALR improves robustness on the union by 8.66%. Additionally, when the initial attack is ℓ_2 and the new attack is Snow (Kaufmann et al., 2019), ALR improves robustness on the

Procedure	Clean	ℓ_2	StAdv	ℓ_∞	Recolor	Avg (known)	Union (known)	Avg (all)	Union (all)
FT Single	80.89	45.45	54.5	6.09	41.98	49.98	41.05	37.0	5.87
FT Single + EWC (0.5)	83.98	58.85	51.15	15.44	51.55	55.00	46.25	44.25	14.54
FT Single + EWC (1)	85.20	57.69	56.18	13.07	50.99	56.93	49.42	44.48	12.69
FT Single + EWC (2)	85.10	57.96	55.14	13.54	51.23	56.55	48.9	44.47	12.99
FT Single + ALR	**87.24**	**62.22**	**61.5**	**21.4**	**70.87**	**61.86**	**55.04**	**54.0**	**21.14**

Table 5: FT Single with EWC compared to FT Single with ALR for the sequence $\ell_2 \rightarrow$StAdv attack (analogous to time step 1 in Table 1). Regularization strength for EWC is shown in parentheses. Accuracy on known attacks are in green cells.

Union of both attacks by 7.85%. We find same trend holds for VR (Appendix J.1).

Random noise based regularization is harmful when used in fine-tuning. Although random noise based regularization can improve robustness when used in the initial training phase of CRT, Figure 3(b) demonstrates that UR in fine-tuning hurts union accuracy for many initial and new attack pairs (corresponding results for GR are present in Appendix J.1). This suggests that while random noise based regularization can be used to perform initial training more efficiently, they should not be used during fine-tuning. Since we found that UR and GR trade off accuracy on the initial attack when used in initial training in §4.3, this suggests that UR and GR generally trade off performance on attacks that are used in training or fine-tuning.

Comparison to EWC. In Table 5, we compare FT Single using EWC (Kirkpatrick et al., 2017) to FT Single with ALR. In these experiments, the model is initially trained on CIFAR-10 to be robust against ℓ_2 attacks, and now we want to finetune to achieve robustness against StAdv attacks (analogous to time step 1 in Table 1). Overall, we find that ALR's improvement in robustness on known and unforeseen attacks is significant compared to EWC. We believe that this is because ALR can also be applied in initial training to boost the initial state of the model prior to finetuning. EWC's improvement over FT Single is similar to using FT Croce results in Table 1 (Time step 1) which uses replay of previous attacks in finetuning.

TAKEAWAY 3. In fine-tuning, adversarial regularization (ALR and VR) can improve Union accuracy significantly (up to $\sim 7\%$) while random noise-based regularization hurts Union accuracy.

5. Discussion and Related Work

This work makes early progress towards deployable defenses that mitigate model obsolescence in the face of evolving adversaries. Such approaches could promote the adoption of robust models, as they allow model trainers to 'patch' against vulnerabilities without training from scratch.

Related Work: Prior works investigate multiattack robustness (MAR) (Maini et al., 2020; Tramèr & Boneh, 2019; Madaan et al., 2020; Croce & Hein, 2020a; Jiang & Singh, 2024) and unforeseen attack robustness (Laidlaw et al., 2021; Zhang et al., 2018; Dai et al., 2022; Jin & Rinard, 2020; Dai et al., 2023). Unlike these methods, we assume that the defender may not know all attacks *a priori* but can adjust their model as new attacks emerge. Croce & Hein (2022) propose a fine-tuning method for MAR on unions of ℓ_p attacks. Our work differs by exploring additional attack types (*e.g.* spatial attacks (Xiao et al., 2018) and color shifts (Laidlaw & Feizi, 2019)) and improvements to the initial training stage prior to fine-tuning. We provide detailed discussion of related work in adversarial ML in Appendix A.

Our problem setting is also related to continual learning (CL). In CL, a set of tasks is learned sequentially with the goal of performing as well as if they were learned simultaneously (Wang et al., 2023a). Few works have studied the intersection of CL and adversarial ML with most works focusing on evaluating or improving the robustness of models trained in the CL framework (Bai et al., 2023; Khan et al., 2022a;b). The most similar to our work is Wang et al. (2023b) which treats different attacks as tasks and uses approaches in CL to sequentially adapt a model against attacks using a different optimization procedure (ie. FGSM or PGD) rather than a different attack type as in our work.

Gradual domain adaptation (Kumar et al., 2020; He et al., 2024; Wang et al., 2022; Zhuang et al., 2024) is another related field which looks adapting a model to distribution shifts with access to intermediate domains with pseudolabels. These intermediate domains can be thought of as gradual shifts in data distribution over time and are not designed adversarially. In comparison, our work looks at changes in the space of attacks over time, and we assume that the defender is able to generate these attacks on their own data, thus ensuring that they have access to labels.

Limitations: More work is needed to improve the performance of regularized CRT, as our approach does not outperform existing baselines in all settings. It also remains unclear whether training from scratch with all attacks or fine-tuning on new attacks is optimal from both a theoretical and empirical perspective. Future work could also compare the convergence rates of training from scratch and CRT. Deriving tighter bounds and potentially better continual robust training methods by bounding the change in loss between the models at each stage remains open. Further limitations and future directions are discussed in Appendix C.

Acknowledgements

We thank Ashwinee Panda and Wenxin Ding for their helpful feedback on the paper. Prateek Mittal is supported by the Princeton SEAS Innovation Grant.

Impact Statement

The defense framework proposed can be useful for safety in practical, high-risk applications of supervised machine learning such as autonomous vehicles (Lab, 2019; Jing et al., 2021; Song et al., 2023), content moderation (Ye et al., 2023; Schaffner et al., 2024), and face authentication (Komkov & Petiushko, 2021; Wei et al., 2022) and provides first steps towards training and updating models in order to maintain robustness over time. However, there are cases in which adversarial examples are used for good (*e.g.* defending against website fingerprinting (Rahman et al., 2020; Shan et al., 2021)) which may be adversely affected by models robust to adversarial examples, including our proposed approach.

References

Bai, T., Chen, C., Lyu, L., Zhao, J., and Wen, B. Towards adversarially robust continual learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

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.

Cianfarani, C., Bhagoji, A. N., Sehwag, V., Zhao, B., Zheng, H., and Mittal, P. Understanding robust learning through the lens of representation similarities. *Advances in Neural Information Processing Systems*, 35:34912–34925, 2022.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. 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. 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

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. Adversarial robustness against multiple and single l_p-threat models via quick fine-tuning of robust classifiers. In *International Conference on Machine Learning*, pp. 4436–4454. PMLR, 2022.

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.

Dai, S., Mahloujifar, S., Xiang, C., Sehwag, V., Chen, P.-Y., and Mittal, P. MultiRobustBench: Benchmarking robustness against multiple attacks. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6760–6785. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dai23c.html.

Dai, S., Ding, W., Bhagoji, A. N., Cullina, D., Zheng, H., Zhao, B., and Mittal, P. Characterizing the optimal $0-1$ loss for multi-class classification with a test-time attacker. *Advances in Neural Information Processing Systems*, 36, 2024a.

Dai, S., Xiang, C., Wu, T., and Mittal, P. Position paper: Beyond robustness against single attack types. *arXiv preprint arXiv:2405.01349*, 2024b.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Ghazanfari, S., Garg, S., Krishnamurthy, P., Khorrami, F., and Araujo, A. R-lpips: An adversarially robust perceptual similarity metric. *arXiv preprint arXiv:2307.15157*, 2023.

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.

Gragnaniello, D., Marra, F., Verdoliva, L., and Poggi, G. Perceptual quality-preserving black-box attack against deep learning image classifiers. *Pattern Recognition Letters*, 147:142–149, 2021. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2021.03.033. URL https://www.sciencedirect.com/science/article/pii/S0167865521001288.

He, Y., Wang, H., Li, B., and Zhao, H. Gradual domain adaptation: Theory and algorithms. *Journal of Machine Learning Research*, 25(361):1–40, 2024.

Howard, J. Imagewang. URL https://github.com/fastai/imagenette/.

Jiang, E. and Singh, G. Ramp: Boosting adversarial robustness against multiple l_p perturbations for universal robustness. *Advances in Neural Information Processing Systems*, 37:43759–43787, 2024.

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

Jing, P., Tang, Q., Du, Y., Xue, L., Luo, X., Wang, T., Nie, S., and Wu, S. Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 3237–3254. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/jing.

Kaufmann, M., Kang, D., Sun, Y., Basart, S., Yin, X., Mazeika, M., Arora, A., Dziedzic, A., Boenisch, F., Brown, T., et al. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

Khan, H., Bouaynaya, N. C., and Rasool, G. Adversarially robust continual learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022a. doi: 10.1109/IJCNN55064.2022.9892970.

Khan, H., Shah, P. M., Zaidi, S. F. A., et al. Susceptibility of continual learning against adversarial attacks. *arXiv preprint arXiv:2207.05225*, 2022b.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Komkov, S. and Petiushko, A. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th international conference on pattern recognition (ICPR)*, pp. 819–826. IEEE, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kumar, A., Ma, T., and Liang, P. Understanding self-training for gradual domain adaptation. In *International conference on machine learning*, pp. 5468–5479. PMLR, 2020.

Lab, T. K. S. Experimental security research of tesla autopilot. *Tencent Keen Security Lab*, 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.

McCloskey, M. and Cohen, N. J. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pp. 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Nern, L. F., Raj, H., Georgi, M., and Sharma, Y. On transfer of adversarial robustness from pretraining to downstream tasks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Rahman, M. S., Imani, M., Mathews, N., and Wright, M. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. *IEEE Transactions on Information Forensics and Security*, 16:1594–1609, 2020.

Rebuffi, S.-A., Croce, F., and Gowal, S. Revisiting adapters with adversarial training. In *The Eleventh International Conference on Learning Representations*.

Schaffner, B., Bhagoji, A. N., Cheng, S., Mei, J., Shen, J. L., Wang, G., Chetty, M., Feamster, N., Lakier, G., and Tan, C. " community guidelines make this the best party on the internet": An in-depth study of online platforms' content moderation policies. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2024.

Shan, S., Bhagoji, A. N., Zheng, H., and Zhao, B. Y. Patch-based defenses against web fingerprinting attacks. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pp. 97–109, 2021.

Song, R., Ozmen, M. O., Kim, H., Muller, R., Celik, Z. B., and Bianchi, A. Discovering adversarial driving maneuvers against autonomous vehicles. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2957–2974, Anaheim, CA, August 2023. USENIX Association. ISBN 978-1-939133-37-3. URL https://www.usenix.org/conference/usenixsecurity23/presentation/song.

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.

Tripuraneni, N., Jordan, M., and Jin, C. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33: 7852–7862, 2020.

Wang, H., Li, B., and Zhao, H. Understanding gradual domain adaptation: Improved analysis, optimal path and beyond. In *International Conference on Machine Learning*, pp. 22784–22801. PMLR, 2022.

Wang, L., Zhang, X., Su, H., and Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023a.

Wang, Q., Liu, Y., Ling, H., Li, Y., Liu, Q., Li, P., Chen, J., Yuille, A., and Yu, N. Continual adversarial defense. *arXiv preprint arXiv:2312.09481*, 2023b.

Watkins, A., Nguyen-Tang, T., Ullah, E., and Arora, R. Adversarially robust multi-task representation learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Wei, X., Guo, Y., and Yu, J. Adversarial sticker: A stealthy attack method in the physical world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (3):2711–2725, 2022.

Wong, E. and Kolter, J. Z. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.

Wong, E., Schmidt, F. R., and Kolter, J. Z. Wasserstein adversarial examples via projected sinkhorn iterations. *CoRR*, abs/1902.07906, 2019. URL http://arxiv.org/abs/1902.07906.

Wu, K., Wang, A., and Yu, Y. Stronger and faster wasserstein adversarial attacks. In *International Conference on Machine Learning*, pp. 10377–10387. PMLR, 2020.

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-.

Ye, Y., Le, T., and Lee, D. Noisyhate: Benchmarking content moderation machine learning models with human-written perturbations online. *arXiv preprint arXiv:2303.10430*, 2023.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In Wilson, R. C., Hancock, E. R., and Smith, W. A. P. (eds.), *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press, 2016. URL http://www.bmva.org/bmvc/2016/papers/paper087/index.html.

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, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D. S., and Hsieh, C. Towards stable and efficient training of verifiably robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skxuk1rFwB.

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.

Zhuang, Z., Zhang, Y., and Wei, Y. Gradual domain adaptation via gradient flow. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iTTZFKrlGV.

This appendix is organized as follows:

1. Additional related work (Appendix A)
2. Applications of CAR (Appendix B)
3. Future directions (Appendix C)
4. Proofs (Appendix D)
5. Connection to variation regularization (Appendix E)
6. Experimental verification of theoretical results (Appendix F)
7. Additional experimental setup details (training and attack parameters, model selection, regularization setup) (Appendix G)
8. Additional experiments
   - Longer attack sequences and different datasets (CIFAR-100 and ImageNette) (Appendix H)
   - Ablations on initial training (comparison to TRADES, attack choice, regularization parameters) (Appendix I)
   - Ablations on fine-tuning (attack choice, regularization parameters) (Appendix J)

A. Additional Related Work

Adversarial Attacks and Defenses: ML models are vulnerable to input-space perturbations known as adversarial examples (Szegedy et al., 2014). These attacks come in different formulations including ℓ_p-norm bounded attacks (Madry et al., 2018; Carlini & Wagner, 2017), spatial transformations (Xiao et al., 2018), color shifts (Laidlaw & Feizi, 2019), JPEG compression and weather changes (Kaufmann et al., 2019), bounded Wasserstein distance (Wong et al., 2019; Wu et al., 2020) as well as attacks based on distances that are more aligned with human perception such as SSIM (Gragnaniello et al., 2021) and LPIPS distances (Laidlaw et al., 2021; Ghazanfari et al., 2023).

Despite the wide variety of attacks that have been introduced, defenses against adversarial examples focus mainly on ℓ_∞ or ℓ_2-norm bounded perturbations (Cohen et al., 2019; Zhang et al., 2020; Madry et al., 2018; Zhang et al., 2019; Croce et al., 2020). Of existing defenses, adversarial training (Madry et al., 2018), an approach that uses adversarial examples generated by the attack of interest during training, can most easily be adjusted to different attacks. In our work, we build off of adversarial training in order to adapt to new adversaries.

Training Techniques for Multi-Robustness:A few prior works have studied the problem of achieving robustness against multiple attacks, under the assumption that all attacks are known a priori. These include training based approaches (Maini et al., 2020; Tramèr & Boneh, 2019; Madaan et al., 2020; Jiang & Singh, 2024) which incorporate adversarial examples from the threat models of interest (usually the combination of ℓ_1, ℓ_2, and ℓ_∞ norm bounded attacks) during training. Croce & Hein (2020a) provides a robustness certificate of all ℓ_p norms given certified robustness against ℓ_∞ and ℓ_1 attacks. Of these approaches (Jiang & Singh, 2024) is similar to ours. Jiang & Singh (2024) looks at the problem of achieving robustness against multiple ℓ_p norms and proposes a logit pairing loss which aims to minimize the KL divergence between the logits of predicting on 2 different ℓ_p attacks. Additionally, they use gradient projection to integrate model updates between natural training and adversarial training for better clean accuracy-robustness tradeoff. In comparison, our work looks at robustness against sequences of attacks including non-ℓ_p attacks. Our regularization term uses ℓ_2 distance between clean and adversarial logits.

Another line of works has looked at defending against attacks that are not known by the defender, which is a problem known as unforeseen robustness. These techniques are all training-based and include Laidlaw et al. (2021) which proposes training based on LPIPS (Zhang et al., 2018), a metric more aligned with human perception than ℓ_p distances, and Dai et al. (2022); Jin & Rinard (2020) which use regularization during training in order to obtain better generalization to unforeseen attacks. Dai et al. (2023) provides a comprehensive leaderboard for the performance of existing defenses against a large variety of attacks at different attack strengths.

Our work differs from these lines of works since we assume that while the defender may not know all attacks a priori, they are allowed to adjust their defense when they become aware of new attacks. The work most similar to ours is Croce

& Hein (2022), which proposes fine-tuning a model robust against one ℓ_p attack to be robust against the union of ℓ_p attacks. Specifically, they demonstrate that we can achieve simultaneous multiattack robustness for the union of ℓ_p attacks by obtaining robustness against ℓ_1 and ℓ_∞ attacks, and thus propose fine-tuning with ℓ_1 and ℓ_∞ attacks to achieve this efficiently. Our work differs from this work since we explore adapting to attacks outside of ℓ_p attacks, investigate ways of improving the initial state of the model prior to fine-tuning, and consider adapting to sequences of attacks.

B. Applications of CAR

Solving CAR is of interest in any safety-critical domain where an attacker is motivated to evade a ML model. A good example is automated content moderation, where malicious actors try to post content that violates policies by uploading obfuscated images . Strategies naturally evolve over time for motivated attackers who can also use numerous open-source methods proposed in the literature, which also evolve over time. Given that ML models will continue to be used in sensitive domains such as finance, cyber-physical systems and medicine, model deployers need methods to update their models to evolving threats.

C. Future Directions

We now discuss a few directions for future work in depth.

Choice of initial attacks and attack similarities. In this work, we looked at ℓ_2 and ℓ_∞ attacks as the initial attack in the CAR problem. However, in practice, we would like to choose an initial attack that is the most representative of the attacks we want to be robust against, in order to generalize to downstream new attacks. Further research on understanding and improving the initial attack can improve the accuracies achieved through training with CRT. Additionally, having ways of measuring attack similarity between the known attacks and new attacks can help allow us to decide whether using CRT is sufficient for achieving good robustness or whether we need to train from scratch or combine the model with other defenses tailored towards the new attack.

Attack Monitoring. One assumption of CAR is that the defender is able to discover when a new attack exists. While this is clear in cases such as a research group publishing a paper with a new attack or a company's security team finding a vulnerabilities, in practice, we would also be interested in recovering after an adversary discovers a new, unknown attack and successfully attacks the model. In this case, we would need a good monitoring system for detecting and synthesizing these new attacks for use with CRT.

Towards real world robustness. In our work, we focus on changes in the defender's knowledge of attacks over time which is useful in cases such as a research or security team discovering a new attack type. A real-time attack setting poses new challenges:

- *No access to threat model*- the defender does not know the threat model and cannot generate adversarial examples. They only have access to the perturbed data generated by the adversary.
- *Missing true labels and no access to the original unperturbed input* - the defender also does not have the corresponding true labels or the original clean input for use in training.
- *Few shot updates* - it becomes critical that the model can be made robust with only a few examples of successful attacks, otherwise it means that the adversary has been exploiting the vulnerabilities of the model for a long time

Defending in this setting is outside of the scope of this paper, but potentially using generative models in order to model the perturbation (Wong & Kolter, 2020) used by the adversary can help to bridge the gap from points (1) and (2) and allow for the defender to apply the attack on their own dataset and finetune with our proposed CRT + ALR. If the generative model is able to learn to model perturbations with only a few adversarial examples, then this can also address (3).

Reducing catastrophic forgetting. In CAR, since attacks are introduced sequentially, catastrophic forgetting is an important problem. In our work, we utilized replay via Croce & Hein (2022)'s fine-tuning approach and also found that ALR reduces catastrophic forgetting to some extent. Future work on reducing catastrophic forgetting can help improve the effectiveness of updating the model with CRT.

Training and fine-tuning efficiency. In our experiments, we combine regularization with Croce & Hein (2022)'s fine-tuning approach due to the effectiveness and efficiency of that approach. Further research on developing better and more efficient fine-tuning techniques for achieving robustness to new attacks (while maintaining robustness against previous attacks) can

improve our CRT framework.

Model capacity. Current works in adversarial robustness literature show that adversarially robust models need higher model capacity (Madry et al., 2018; Gowal et al., 2020; Cianfarani et al., 2022). As we increase the space of attacks to defend against, we may need to increase the capacity of the model in order to achieve multi-robustness (Dai et al., 2024a). An interesting future direction is looking at the connection between model capacity and CAR and seeing if adding more parameters to the network during fine-tuning (such as using adapters (Rebuffi et al.)) can be used to address the issue of model capacity.

Theory. We believe further work is necessary to extend the theory of CAR. Our results focus on the relationship between robust loss and logit distance between attacks for a *single model*. However, we do not extend them to comparisons between loss under different attacks for *different* models, such as the initial robust model and the one at the end of fine-tuning. Additionally, the CAR framework could be extended to the multi-task setting, as is the case in multi-task representation learning (Watkins et al., 2024; Tripuraneni et al., 2020). These prior works connect the ability of a class of models to learn a set of tasks to the complexity of that class (measured using Gaussian or Rademacher complexity, for example). Similar methods may also be useful for proving a model's ability to defend against multiple adversaries.

D. Proofs

D.1. Proof of Theorem 3.1

The proof of Theorem 3.1 adapts that of Theorem I from Nern et al. (2023) by considering multiple attacks compared to the single one considered there.

Proof. Define independent random variables $D_1, \dots, D_n$ as

$$D_i = \max_{x_i' \in C_1(x_i)} \ell(h(x_i'), y_i) - \max_{x_i'' \in C_2(x_i)} \ell(h(x_i''), y_i),$$

based on independently drawn data points with probability distribution $\mathcal{P}(X)$. Using Hoeffding's inequality, we get

$$\begin{aligned} &\mathbb{P}\left(\left|\sum_{i=1}^n D_i - n\mathbb{E}[D]\right| \geq t\right) \leq 2 \cdot \exp\left(\frac{-2t^2}{nM_2^2}\right) \\ \implies &\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^n D_i - \mathbb{E}[D]\right| \leq M_2\sqrt{\frac{\log(\rho/2)}{-2n}}\right) \geq 1-\rho. \end{aligned}$$

Thus, with probability at least $1-\rho$ it holds that

$$\begin{aligned} \mathbb{E}[D] &= |\mathcal{L}_1(h) - \mathcal{L}_2(h)| \\ &= \left|\mathbb{E}_{(x,y)}\left[\max_{x' \in C_1(x)} \ell(h(x'), y) - \max_{x'' \in C_2(x)} \ell(h(x''), y)\right]\right| \\ &\leq \left|\frac{1}{n}\sum_{i=1}^n \max_{x' \in C_1(x)} \ell(h(x'), y_i) - \max_{x'' \in C_2(x)} \ell(h(x''), y_i)\right| + M_2\sqrt{\frac{\log(\rho/2)}{-2n}}. \end{aligned} \tag{1}$$

We can further bound the first term on the right hand side, since the loss function $\ell(r, y)$ is M_1-Lipschitz in $\|\cdot\|_2$ for $r \in h(X)$:

$$\begin{aligned} &\left|\frac{1}{n}\sum_{i=1}^n \max_{x' \in C_1(x)} \ell(h(x'), y_i) - \max_{x'' \in C_2(x)} \ell(h(x''), y_i)\right| \\ &\leq \left|\frac{1}{n}\sum_{i=1}^n |\ell(h(x_i'), y_i) - \ell(h(x_i''), y_i)|\right| \\ &\leq M_1 \frac{1}{n}\sum_{i=1}^n \|h(x_i') - h(x_i'')\|_2, \end{aligned} \tag{2}$$

where $x'_1, \ldots, x'_n$ with $x'_i \in C_1(x_i)$ and $x''_1, \ldots, x''_n$ with $x''_i \in C_2(x_i)$ are chosen to maximize $\ell(h(\cdot), y_i)$ for each i. The perturbed samples represented in this inequality might not maximize the distance between the logits, but that distance can be bounded by the maximally distant perturbations within each neighborhood. Making use of the triangle inequality, we obtain:

$$\begin{aligned}
&\sum_{i=1}^{n} \|h(x'_i) - h(x''_i)\|_2 \\
&= \sum_{i=1}^{n} \|(h(x'_i) - h(x_i)) - (h(x''_i) - h(x_i))\|_2 \\
&\leq \sum_{i=1}^{n} \|h(x'_i) - h(x_i)\|_2 + \|h(x''_i) - h(x_i)\|_2 \\
&\leq \sum_{i=1}^{n} \max_{x' \in C_1(x_i)} \|h(x') - h(x_i)\|_2 + \max_{x'' \in C_2(x_i)} \|h(x'') - h(x_i)\|_2. \qquad (3)
\end{aligned}$$

We then achieve our final result, recalling the assumption that $\mathcal{L}_1(h) \geq \mathcal{L}_2(h)$:

$$\begin{aligned}
\mathcal{L}_1(h) - \mathcal{L}_2(h) &= |\mathcal{L}_1(h) - \mathcal{L}_2(h)| \\
&\leq M_1 \frac{1}{n} \sum_{i=1}^{n} \Big(\max_{x' \in C_1(x_i)} \|h(x') - h(x_i)\|_2 + \max_{x'' \in C_2(x_i)} \|h(x'') - h(x_i)\|_2 \Big) + D, \qquad (4)
\end{aligned}$$

where $D = M_2 \sqrt{\frac{\log(\rho/2)}{-2n}}$. □

D.2. Proof of Corollary 3.2

Proof. Define independent random variables $D_1, \ldots, D_n$ as

$$D_i = \max_{x'_i \in C_1(x_i) \cup C_2(x_i)} \ell(h(x'_i), y_i) - \ell(h(x_i), y_i),$$

based on independently drawn data points with probability distribution $\mathcal{P}(\mathcal{X})$. Using Hoeffding's inequality, we get

$$\begin{aligned}
&\mathbb{P}\left(\left| \sum_{i=1}^{n} D_i - n\mathbb{E}[D] \right| \geq t \right) \leq 2 \cdot \exp\left(\frac{-2t^2}{nM_2^2} \right) \\
\implies &\mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^{n} D_i - \mathbb{E}[D] \right| \leq M_2 \sqrt{\frac{\log(\rho/2)}{-2n}} \right) \geq 1 - \rho.
\end{aligned}$$

Thus, with probability at least $1 - \rho$ it holds that

$$\begin{aligned}
\mathbb{E}[D] &= |\mathcal{L}_{1,2}(h) - \mathcal{L}(h)| \\
&= \left| \mathbb{E}_{(x,y)} \left[\max_{x' \in C_1(x) \cup C_2(x)} \ell(h(x'), y) - \ell(h(x), y) \right] \right| \\
&\leq \left| \frac{1}{n} \sum_{i=1}^{n} \max_{x' \in C_1(x_i) \cup C_2(x_i)} \ell(h(x'), y_i) - \ell(h(x_i), y_i) \right| + M_2 \sqrt{\frac{\log(\rho/2)}{-2n}}. \qquad (5)
\end{aligned}$$

We can further bound the first term on the right hand side, since the loss function $\ell(r, y)$ is M_1-Lipschitz in $\|\cdot\|_2$ for $r \in h(\mathcal{X})$:

$$\begin{aligned}
&\left|\frac{1}{n}\sum_{i=1}^{n}\max_{x'\in C_1(x_i)\cup C_2(x_i)}\ell(h(x'),y_i)-\ell(h(x_i),y_i)\right| \\
&=\left|\frac{1}{n}\sum_{i=1}^{n}|\ell(h(x_i'),y_i)-\ell(h(x_i),y_i)|\right| \\
&\leq M_1\frac{1}{n}\sum_{i=1}^{n}\|h(x_i')-h(x_i)\|_2,
\end{aligned} \tag{6}$$

where $x_1', \ldots, x_n'$ with $x_i' \in C_1(x_i) \cup C_2(x_i)$ are chosen to maximize $\ell(h(\cdot), y_i)$ for each i. The perturbed samples represented in this inequality might not maximize the distance between the logits, but that distance can be bounded by the maximally distant perturbations within each neighborhood.

$$\begin{aligned}
&\sum_{i=1}^{n}\|h(x_i')-h(x_i)\|_2 \\
&\leq\sum_{i=1}^{n}\max_{x'\in C_1(x_i)\cup C_2(x_i)}\|h(x')-h(x_i)\|_2 \\
&=\sum_{i=1}^{n}\max_{C\in\{C_1,C_2\}}\max_{x'\in C(x_i)}\|h(x')-h(x_i)\|_2
\end{aligned} \tag{7}$$

We then achieve our final result :

$$\begin{aligned}
\mathcal{L}_{1,2}(h)-\mathcal{L}(h)&=|\mathcal{L}_{1,2}(h)-\mathcal{L}(h)| \\
&\leq M_1\frac{1}{n}\sum_{i=1}^{n}\Big(\max_{x'\in C_1(x_i)}\|h(x')-h(x_i)\|_2+\max_{x''\in C_2(x_i)}\|h(x'')-h(x_i)\|_2\Big)+D,
\end{aligned} \tag{8}$$

where $D = M_2\sqrt{\frac{\log(\rho/2)}{-2n}}$. □

D.3. Relating the Loss Gap to Internal Representations

While our results bound the robust loss gap in terms of the distance between logits of samples perturbed with different attacks, similar results hold for the distance between internal activations. To show how our results can apply to common transfer learning settings (such as that of Nern et al. (2023)), we prove the following corollary:

Corollary D.1. *Let $h : \mathbb{R}^d \to \mathbb{R}^c$ be a c class neural network classification model with a final linear layer (i.e. $h(c) = Wg(x)$, where $g : \mathbb{R}^d \to \mathbb{R}^r$, and $W \in \mathbb{R}^{c\times r}$). Assume that loss $\ell(\hat{y}, y)$ is M_1-Lipschitz in $\|\cdot\|_\alpha$ for $\alpha \in \{1, 2, \infty\}$, for $\hat{y} \in h(X)$ with $M_1 > 0$ and bounded by $M_2 > 0$, i.e. $0 \leq \ell(\hat{y}, y) \leq M_2\ \forall \hat{y} \in h(X)$. Then, for a subset $\mathbb{X} = \{x_i\}_{i=1}^n$ independently drawn from $\mathcal{D}$, the following holds with probability at least $1 - \rho$:*

$$\mathcal{L}_1(h)-\mathcal{L}_2(h)\leq L_\alpha(W)M_1\frac{1}{n}\sum_{i=1}^{n}\left(\max_{x'\in C_1(x_i)}\|g(x')-g(x_i)\|_2+\max_{x'\in C_2(x_i)}\|g(x')-g(x_i)\|_2\right)+D,$$

where $D = M_2\sqrt{\frac{\log(\rho/2)}{-2n}}$ and

$$L_\alpha(W):=\begin{cases}\|W\|_2 & ,\text{if } \|\cdot\|_\alpha=\|\cdot\|_2,\\ \sum_i\|W_i\|_2 & ,\text{if } \|\cdot\|_\alpha=\|\cdot\|_1,\\ \max_i\|W_i\|_2 & ,\text{if } \|\cdot\|_\alpha=\|\cdot\|_\infty.\end{cases}$$

Proof. From (5), (6), and the definition of h, we have that

$$|\mathcal{L}_1 - \mathcal{L}_2| \leq M_1 \frac{1}{n} \sum_{i=1}^{n} \|Wg(x_i') - h(x_i'')\|_2 + M_2 \sqrt{\frac{\log \rho/2}{-2n}}.$$

We then apply Lemma 2 from Nern et al. (2023) and the definition of L_α:

$$\begin{aligned} & M_1 \frac{1}{n} \sum_{i=1}^{n} \|Wg(x_i') - Wg(x_i'')\|_\alpha \\ & \qquad \leq L_\alpha(W) M_1 \frac{1}{n} \sum_{i=1}^{n} \|g(x_i') - g(x_i'')\|_2. \end{aligned}$$

As in the proof for Theorem 3.1, the perturbed samples represented in this inequality might not maximize the distance between the representations, but that distance can be bounded by the maximally distant perturbations within each neighborhood. Making use of the triangle inequality, we obtain:

$$\begin{aligned} & \sum_{i=1}^{n} \|g(x_i') - g(x_i'')\|_2 \\ & = \sum_{i=1}^{n} \|(g(x_i') - g(x_i)) - (g(x_i'') - g(x_i))\|_2 \\ & \leq \sum_{i=1}^{n} \|g(x_i') - g(x_i)\|_2 + \|g(x_i'') - g(x_i)\|_2 \\ & \leq \sum_{i=1}^{n} \max_{x' \in C_1(x_i)} \|g(x') - g(x_i)\|_2 + \max_{x'' \in C_2(x_i)} \|g(x'') - g(x_i)\|_2. \end{aligned}$$

We then achieve our final result:

$$\begin{aligned} \mathcal{L}_1(h) - \mathcal{L}_2(h) &= |\mathcal{L}_1(h) - \mathcal{L}_2(h)| \\ & \leq L_\alpha(W) M_1 \frac{1}{n} \sum_{i=1}^{n} \Big(\max_{x' \in C_1(x_i)} \|g(x') - g(x_i)\|_2 + \max_{x'' \in C_2(x_i)} \|g(x'') - g(x_i)\|_2 \Big) + D, \end{aligned}$$

where $D = M_2 \sqrt{\frac{\log(\rho/2)}{-2n}}$. □

E. Connection Between Adversarial ℓ_2 Regularization and Variation Regularization

In this section, we will show the relationship between adversarial ℓ_2 regularization (ALR) and variation regularization (VR) (Dai et al., 2022). To begin, we first revisit the definitions of ALR and VR:

$$R_{\text{ALR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \max_{x' \in C(x_i)} \|h(x') - h(x_i)\|_2$$

$$R_{\text{VR}}(h, K(t)) = \frac{1}{m} \sum_{i=1}^{m} \max_{x', x'' \in C(x_i)} \|h(x') - h(x'')\|_2$$

Since VR optimizes over 2 perturbations x' and x'' for each example while ALR optimizes only for x', it is clear that $R_{\text{ALR}} \leq R_{\text{VR}}$. Additionally, we note that:

$$\begin{aligned} R_{\text{VR}}(h, K(t)) &= \frac{1}{m} \sum_{i=1}^{m} \max_{x', x'' \in C(x_i)} \|h(x') - h(x) + h(x) - h(x'')\|_2 \\ & \leq \frac{1}{m} \sum_{i=1}^{m} \max_{x', x'' \in C(x_i)} \|h(x') - h(x)\|_2 + \|h(x) - h(x'')\|_2 \end{aligned}$$

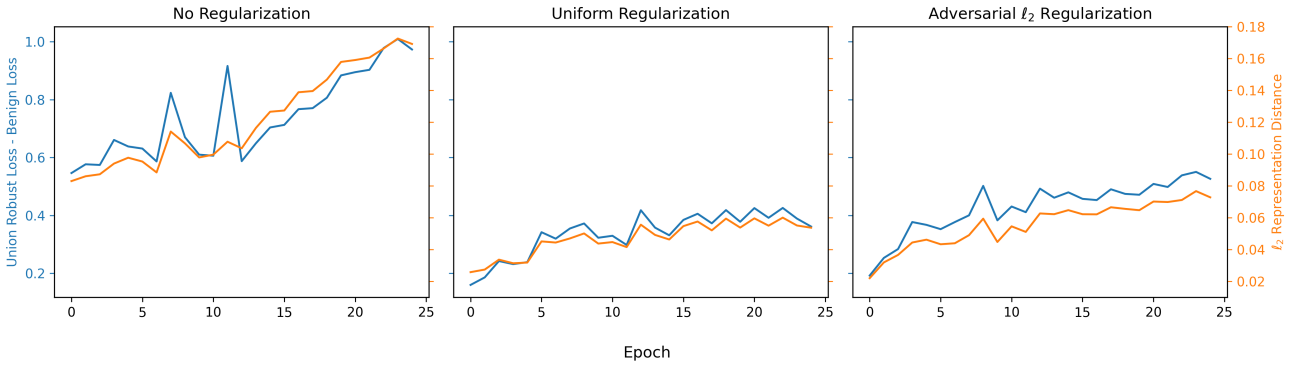


Figure 4: Adversarial loss gap ($\mathcal{L}_{1,2}(h) - \mathcal{L}(h)$) and average ℓ_2 distance between logits of ℓ_2 ($\epsilon = 0.5$, representing P_{C_1}) and StAdv ($\epsilon = 0.05$, representing P_{C_2}) attacked samples over 25 epochs of fine-tuning using (Croce & Hein, 2022)'s fine-tuning method, both with and without regularization. Each model is fine-tuned starting from a model that is adversarially trained against an ℓ_2 adversary, as described in Section 4.1. In all training scenarios, there is a visible correlation between the loss gap and the logit distance, aligning with the theoretical result in Corollary 3.2.

$$= \frac{2}{m} \sum_{i=1}^{m} \max_{x' \in C(x_i)} \|h(x') - h(x)\|_2$$

$$= 2R_{\text{ALR}}$$

Thus, ALR and VR are related in the sense that $R_{\text{ALR}} \leq R_{\text{VR}} \leq 2R_{\text{ALR}}$.

F. Experimental Verification of Theoretical Results

We now briefly demonstrate that our chosen regularization terms align with our theoretical results. In Figure 4, we start with WRN-28-10 models that were adversarially trained to be robust against ℓ_2-bounded attacks, and fine-tune them to increase their robustness against StAdv attacks using either no regularization, uniform regularization, or adversarial ℓ_2 regularization. We observe a number of trends:

Sensitivity correlates with loss gap. Whether or not regularization is used, there is a clear correlation between total adversarial sensitivity across both attacks (i.e. $\max_{x' \in C_1(x)} \|h(x') - h(x)\| + \max_{x' \in C_2(x)} \|h(x') - h(x)\|$) and the loss gap between the union robust loss and the benign loss (i.e. $\mathcal{L}_{1,2}(h) - \mathcal{L}(h)$).

Regularization reduces sensitivity and loss gap. Both metrics are significantly lower throughout fine-tuning when regularization is used, indicating that regularization is successfully targeting our theoretical bounds.

Loss gap increases over time. Across all three models there is an increase in both loss gap and adversarial sensitivity over the course of fine-tuning. While this may seem like a failure of regularization, the benefit is more apparent when further analyzing what is causing the loss gap to increase. In the regularized fine-tuning runs, both benign and robust losses are decreasing, with benign loss decreasing more quickly. This is likely influenced by an initial increase in benign loss at the very beginning of fine-tuning which is not captured in Figure 4. However, without regularization, benign loss decreases while union robust loss increases. This shows us that despite theoretically targeting the gap between union robust loss and benign loss, the use of regularization still aids in individually reducing both losses in absolute terms.

G. Additional Experimental Setup Details

Additional training details. For initial training, we start with a learning rate of 0.1 and then use the multistep learning rate scheduling proposed by Gowal et al. (2020); specifically, we scale the learning rate down by a factor of 10 halfway and 3/4 of the way through initial training or fine-tuning. For fine-tuning, we maintain a learning rate of 0.001. We train with SGD with momentum of 0.9 and weight decay of 0.0005.

Additional Attack parameters in training. Following other works on adversarial robustness, we use a step size of 0.075 for ℓ_2 attacks on CIFAR-10, 0.15 for ℓ_2 attacks on ImageNette, and $\frac{2}{255}$ for ℓ_∞ attacks. For other attacks, we use $\frac{\epsilon}{8}$ where ϵ is the attack strength as the step size during training. We provide visualizations of each perturbation type in Figure 5.

(a) Clean image (b) Most commonly used perturbation size (c) Largest perturbations used

Figure 5: **Attack visualizations for CIFAR-10.** The original image is portrayed in (a), the perturbed images at perturbation budgets used for most evaluations is shown in (b), and the largest perturbation sizes used in evaluations (diagonal entries in Figure 3) are shown in (c).

Model selection. In the main paper, we stated that we perform evaluation using the epoch at which the model has the best performance measured across known attack types. Specifically, after each epoch of training, we evaluate the performance of each model against the attacks used during training (with the same attack parameters as used during training). For training with AVG, we use the best performing model with respect to the AVG objective (which is the model with the best performance measured as an average over individual attack accuracies). Meanwhile for MAX and FT MAX, we use the best performing model with respect to the MAX objective (which is the best performing model across the union of all attacks). For procedures that only use a single attack per batch during training (Random, FT Single, FT Croce, and our procedure), we use the best performing model measured by sampling attacks per batch randomly.

Regularization setup. We note that all attacks used in this paper use a gradient based optimization scheme for finding the attack. In order to compute regularization for non-ℓ_p threat models, we follow the same optimization scheme used by the attack (Xiao et al., 2018; Laidlaw & Feizi, 2019; Kaufmann et al., 2019) but replace the classification loss portion of the optimization objective to be the ℓ_2 distance between features/logits between the perturbed and unperturbed input. For fine-tuning with regularization, since Croce & Hein (2022)'s fine-tuning approach only uses a single attack per batch, we structure the regularization to mimic Croce & Hein (2022)'s fine-tuning procedure. Specifically, for each batch, the regularization is for a single attack type (the same one which is selected to use with adversarial training by Croce & Hein (2022)'s fine-tuning approach). This helps to reduce the overhead from regularization.

H. Additional Experimental Results for CAR

H.1. Addition Datasets and Attack Sequences

We present additional results for CAR on CIFAR-10 in Tables 6 and 7, results for CAR on Imagenette in Table 8 and 9 and results for CIFAR-100 in Tables 10 and 11. We also compare different fine-tuning approaches in the absense of regularization.

Training time and robust performance. We find that fine-tuning with MAX objective (FT MAX) or Croce & Hein (2022) (FT Croce) can generally achieve robustness across previous attacks and the new attack in the sequence comparable to training from scratch. For example, in Table 7, when fine-tuning to gain robustness against StAdv attack starting from a model initially trained with adversarial training on ℓ_∞ attacks on CIFAR-10, we find that FT MAX achieves 50.75% average robustness across the two attacks and 41.57% union robustness across the two attacks, and FT Croce achieves 49.48% average robustness and 29.69% union robustness. These values lie within (or even above) the range obtained through training from scratch (42.23%-49.61% average robustness and 28.03%-40.8% union robustness). We find that this trend generally holds as well across time steps when new attacks are introduced, when using a different sequence ordering (Table 6).

Of these two techniques, we find that FT MAX generally achieves higher average and union accuracies across the set of known attacks, but is less efficient when used in fine-tuning. For example, In Table 7, FT MAX takes 3.99 hours for 10 epochs of fine-tuning from an ℓ_∞ robust model while FT Croce takes 2.31 hours. The time complexity of FT MAX also scales as the number of attacks increases, leading to 7.9 hours of fine-tuning for 10 epochs when there are 4 known attacks while FT Croce maintains approximately the same training time.

In comparison to naively training from scratch, we also find that these fine-tuning techniques can be much more efficient. For example, a model robust to a sequence of 4 attacks in Table 6 can be found in roughly 17 hours using CRT, but training from scratch each time would require 44 hours cumulatively.

Importance of replay. We find that replay of previous attacks is important for achieving good robustness across the set of known attacks when training with CRT. Fine-tuning with only the new attack (FT Single) usually leads to rapid forgetting of the previous attack. For example, in Table 7 we observe that the accuracy of robustness on the initial attack (ℓ_∞) drops to 31.14% robust accuracy at time step 1 (from the initial accuracy of 51.49% at time step 0) and then further drops to 25.27% at time step 2 when the third attack (Recolor) is introduced. This forgetting is independent from tradeoffs between attacks as we find that training from scratch and FT MAX and FT Croce techniques can all achieve at least 40% ℓ_∞ accuracy at time step 1 and at least 35% ℓ_∞ accuracy at time step 2. The forgetting of previous attacks is also analogous to catastrophic forgetting of previous tasks in continual learning (Wang et al., 2023b; McCloskey & Cohen, 1989). We note however that forgetting is less of a limitation in CAR than in continual learning since the defender's knowledge set only grows over time; they do not forget the formulation of previous attacks and can thus can always use methods such as replay.

ALR applied on logits vs features. In Table 6, we also provide results for using regularization based on distances in the feature space (before the final linear layer), which are labelled with "+ ALR feature". Overall we observe that using regularization in the feature space can also help improve performance on average and union robustness across known attacks as well as improve unforeseen robustness over baselines. However, we observe that feature space regularization leads to larger tradeoffs in clean accuracy than regularization on the logits ("+ ALR" rows) while robust performance is comparable to regularization applied on the logits.

Training durations. Across all tables we also provide experiments for fine-tuning with 25 epochs (as opposed to 10 epochs reported in the main body). We find that increasing the number of fine-tuning epochs can help methods such as FT Croce achieve robustness closer to that of training from scratch, but at the cost of increased time for updating the model.

Performance on other datasets. We find that the gain in performance through using ALR varies across datasets. For Imagenette the gain in performance is generally much smaller than on CIFAR-10 (ALR closes the gap between fine-tuning based updates and training from scratch rather than surpassing training from scratch as in CIFAR-10. On CIFAR-100 ALR generally does not improve performance over fine-tuning. We believe that this is because achieving robustness on multiple attacks is quite hard on CIFAR-10; clean accuracy is between 60-70% and robust accuracies are even lower with StAdv and ℓ_∞ robustness only achieving up to 32% robust accuracy and 25% robust accuracy respectively.

I. Initial Training Ablations

In this section, we present some ablations across regularization strength of each regularization method on the initial training portion of our approach pipeline. We present ablation results for CIFAR-10 and ImageNette.

I.1. Impact of random noise parameter σ

To investigate the impact of the noise parameter σ, we perform initial training on CIFAR-10 with uniform and gaussian regularization at different values of σ. We maintain a value of regularization strength $\lambda = 5$ to isolate the impact of the

Time Step	Procedure	Threat Models	Clean	ℓ_2	StAdv	ℓ_∞	Recolor	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	AT	ℓ_2	**91.17**	69.7	2.08	28.41	44.94	69.7	69.7	36.28	1.24	8.35
	AT + ALR ($\lambda = 1$)	ℓ_2	89.43	**69.84**	**48.23**	**34.00**	**65.46**	**69.84**	**69.84**	**54.38**	**31.27**	11.15
	AT + ALR feature ($\lambda = 5$)	ℓ_2	83.7	63.1	26.57	31.6	62.53	63.1	63.1	45.95	20.16	11.13
1	AVG	ℓ_2, StAdv	87.74	62.17	50.92	17.17	45.47	56.55	47.55	43.93	15.92	23.72
	MAX	ℓ_2, StAdv	86.18	58.65	57.21	11.21	43.07	57.93	51.72	42.54	11.03	23.69
	Random	ℓ_2, StAdv	84.91	57.77	59.74	14.05	44.88	58.76	52.15	44.11	13.68	10.92
	FT MAX (10 ep)	ℓ_2, StAdv	83.73	57.07	58.67	12.51	49.03	57.87	51.32	44.32	12.36	4
	FT MAX (25 ep)	ℓ_2, StAdv	84.85	56.44	61.34	10.35	48.08	58.89	52.52	44.05	10.24	10
	FT Croce (10 ep)	ℓ_2, StAdv	84.7	57.88	54.27	14.38	51.08	56.07	48.13	44.4	13.8	2.4
	FT Croce (25 ep)	ℓ_2, StAdv	86.24	58.94	57.37	13.26	50.36	58.16	50.89	44.98	13	5.98
	FT Single (10 ep)	ℓ_2, StAdv	80.89	45.45	54.5	6.09	41.98	49.98	41.05	37	5.87	2.78
	FT Single (25 ep)	ℓ_2, StAdv	81.21	44.17	54.6	5.56	40.95	49.38	39.76	36.32	5.36	6.92
	FT Single + ALR (10 ep)	ℓ_2, StAdv	87.24	62.22	61.5	21.4	**70.87**	61.86	55.04	54	21.14	4.24
	FT Single + ALR (25 ep)	ℓ_2, StAdv	87.54	61.21	60.38	20.81	69.49	60.8	54.22	52.97	20.48	8.77
	FT Single + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv	81.79	56.98	60.28	20.59	63.64	58.63	51.65	50.37	20.21	3.52
	FT Single + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv	81.26	60.43	57.61	**28.17**	67.95	59.02	51.99	53.54	**27.24**	3.53
	FT Croce + ALR (10 ep)	ℓ_2, StAdv	86.03	59.18	**65.14**	15.36	63.31	**62.16**	**55.83**	50.75	15.29	3.47
	FT Croce + ALR (25 ep)	ℓ_2, StAdv	**88.5**	**64.88**	58.98	23.9	70.79	61.93	55.03	**54.64**	23.33	7.96
	FT Croce + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv	83.19	61.28	59.04	23.98	62.69	60.16	53.25	51.75	23.2	2.97
	FT Croce + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv	83.51	61.69	61.76	23.33	62.48	61.73	55.25	52.31	22.77	3.13
2	AVG	ℓ_2, StAdv, ℓ_∞	85.98	67.60	45.81	42.39	62.43	51.93	34.05	54.56	33.39	33.12
	MAX	ℓ_2, StAdv, ℓ_∞	84.54	54.87	52.33	38.23	55.90	48.48	35.25	50.33	34.08	79.04
	Random	ℓ_2, StAdv, ℓ_∞	39.52	67.46	47.35	42.12	63.61	52.31	35.46	55.13	34.79	10.92
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞	83.16	65.63	56.68	36.9	65.69	53.07	35.18	56.23	34.83	5.62
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞	83.99	65.69	58.16	37.21	65.52	53.69	35.76	56.65	35.31	12.88
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞	85.05	67.3	48.07	33.38	62.52	49.58	28.96	52.82	28.63	2.27
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞	86.14	67.3	52.47	35.86	63.43	51.88	32.54	54.77	32.08	5.01
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞	87.99	**70.53**	11.17	41.63	63.46	41.11	7.95	46.7	7.74	1.57
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞	88.67	70.23	8.79	43.4	63.03	40.81	6.19	46.36	6.05	3.91
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	**88.74**	69.15	47.33	42.08	68.62	52.85	36.66	56.8	36.62	2.26
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	88.14	68.26	49.1	41.48	66.73	52.95	**37.55**	56.39	37.5	5.4
	FT Single + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv, ℓ_∞	85.69	67.62	29.42	43.68	68.75	46.91	24.44	52.37	24.38	2.16
	FT Single + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv, ℓ_∞	84.03	67.64	42.03	**44.36**	71.36	51.34	32.54	56.35	32.48	2.29
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	86.57	67.99	**61.55**	36.59	72.16	**55.38**	35.68	59.57	35.52	2.87
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	86.96	68.91	57.21	39.65	**72.22**	55.26	37.25	**59.5**	37.14	6.87
	FT Croce + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv, ℓ_∞	83.13	66.91	56.76	38.66	68.57	54.11	35.95	57.73	35.76	2.82
	FT Croce + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv, ℓ_∞	84.25	68.14	57.7	39.8	70.29	55.21	37.4	58.98	**37.21**	2.79
3	AVG	ℓ_2, StAdv, ℓ_∞, Recolor	87.77	68.55	39.55	**41.97**	67.93	54.5	30.39	54.5	30.39	50.54
	MAX	ℓ_2, StAdv, ℓ_∞, Recolor	84.3	57.62	52.3	41.69	65.1	54.18	**37.44**	54.18	**37.44**	55.54
	Random	ℓ_2, StAdv, ℓ_∞, Recolor	86.32	65.87	47.82	35.04	68.35	54.27	30.76	54.27	30.76	12.41
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	83.64	66.21	57.53	37.77	69.32	57.71	36.02	57.71	36.02	8.45
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	83.9	65.72	57.84	38.37	68.84	57.69	36.87	57.69	36.87	21.44
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	86.64	**68.76**	44.81	36.02	68.05	54.41	29.44	54.41	29.44	2.34
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	87.11	67.89	49.57	35.58	67.05	55.02	31.21	55.02	31.21	5.9
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	90.41	66.47	3.93	29.6	69.03	42.26	2.49	42.26	2.49	3.11
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	**90.89**	65.14	3.02	30.32	68.54	41.75	1.92	41.75	1.92	7.41
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	90.45	61.58	25.77	27.43	69.26	46.01	19.2	46.01	19.2	4.24
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	90.4	57.07	24.91	22.91	67.39	43.07	17.21	43.07	17.21	9.79
	FT Single + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	90.15	57.89	8.75	22.86	72.27	40.44	6.61	40.44	6.61	3.94
	FT Single + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	88.44	66.03	18.88	34.17	69.35	47.11	16.1	47.11	16.1	3.76
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	87.62	68.14	58.5	36.39	72.35	58.85	34.92	58.85	34.92	3.35
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	87.05	68.05	**59.26**	38.38	**73.42**	**59.78**	36.83	**59.78**	36.83	7.78
	FT Croce + ALR feature ($\lambda = 2$, 10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	84.78	67.67	53.13	40.25	69.99	57.76	36.3	57.76	36.3	3.04
	FT Croce + ALR feature ($\lambda = 5$, 10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	83.94	67.28	59.21	39.38	71.67	59.38	37.15	59.38	37.15	2.91

Table 6: **Continual Robust Training on CIFAR-10 (Sequence of 4 attacks starting with ℓ_2).** The learner initially has knowledge of ℓ_2 attacks and over time, we are sequentially introduced to StAdv, ℓ_∞, and ReColor attacks. We report clean accuracy, accuracy on different attack types, and average and union accuracies. The threat models column represents the set of attacks known to the defender and accuracies on known attacks are highlighted with in green cells. "FT" procedures are fine-tuning approaches starting from adversarially trained to ℓ_2 model (AT) and then sequentially fine-tuning with new attacks for 25 epochs. AVG, MAX, and Random strategies train models from scratch with all attacks for 100 epochs. The "Avg (known)" and "Union (known)" columns represent average and union accuracies on attacks known to the defender while "Avg (all)" and "Union (all)" columns represent average and union accuracies on all four attacks. Additionally, we report training times for the procedure (non-cumulative) in the "Time" column. Best performance out of both training from scratch and fine-tuning are bolded, while best performance when only comparing fine-tuning approaches is underlined.

Time Step	Procedure	Threat Models	Clean	ℓ_∞	StAdv	Recolor	ℓ_2	Avg (known)	Union (known)	Avg (all)	Union (all)	Time
0	AT	ℓ_∞	**85.93**	51.44	14.87	62.48	**59.48**	51.44	51.44	47.07	11.9	7.52
	AT + ALR	ℓ_∞	83.18	**51.49**	**34.78**	**58.15**	58.15	**51.49**	**51.49**	**53.27**	**29.87**	11.12
1	AVG	ℓ_∞, StAdv	**86.44**	30.05	54.4	46.71	52.1	42.23	28.03	45.81	26.75	23.68
	MAX	ℓ_∞, StAdv	82.62	44.96	53.68	64.24	**60.85**	49.32	40.8	55.93	39.81	23.68
	Random	ℓ_∞, StAdv	83.15	40.86	58.37	60.53	58.17	49.61	38.95	54.48	37.64	11.70
	FT MAX (10 ep)	ℓ_∞, StAdv	81.63	44.13	57.38	66.66	60.27	50.75	41.57	57.11	40.96	3.99
	FT MAX (25 ep)	ℓ_∞, StAdv	81.99	44.32	57.8	66.25	60.29	51.06	41.98	57.16	41.25	9.93
	FT Croce (10 ep)	ℓ_∞, StAdv	82.66	44.75	54.2	65.99	60.27	49.48	39.69	56.3	39.01	2.31
	FT Croce (25 ep)	ℓ_∞, StAdv	83.55	45.12	53.25	66.44	60.65	49.19	39.43	56.36	38.74	5.44
	FT Single (10 ep)	ℓ_∞, StAdv	80.39	31.14	55.88	59.13	51.58	43.51	29.01	49.43	28.67	2.77
	FT Single (25 ep)	ℓ_∞, StAdv	79.85	31.34	54.86	58.69	51.43	43.1	29.01	49.08	28.66	6.6
	FT Single + ALR (10 ep)	ℓ_∞, StAdv	82.77	35.67	57.92	68.38	54.91	46.8	33.69	54.22	33.65	3.51
	FT Single + ALR (25 ep)	ℓ_∞, StAdv	81.81	35.4	59.47	68.63	54.34	47.44	33.72	54.46	33.66	8.73
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv	82.94	**46.39**	**64.13**	**73.58**	59.41	**55.26**	**44.47**	**60.88**	**44.03**	2.99
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv	82.3	45.89	63.76	72.8	59.56	54.82	44	60.5	43.54	7.5
2	AVG	ℓ_∞, StAdv, Recolor	88.67	39.46	47.1	66.87	57.16	51.14	32.61	52.65	32.55	39.72
	MAX	ℓ_∞, StAdv, Recolor	83.42	44.54	53.06	67.56	60.71	55.05	40.23	56.47	40.17	47.21
	Random	ℓ_∞, StAdv, Recolor	83.23	35.01	54.7	68.68	**62.92**	52.8	32.83	55.33	32.83	13.81
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor	81.97	44.1	57.36	68.68	60.37	56.71	41.21	57.63	41.2	6.72
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor	82.24	44.36	58.52	68.87	60.23	57.25	**41.73**	57.99	**41.67**	16.69
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor	84.98	43.32	52.45	69.46	61.04	55.08	37.05	56.57	37.01	2.53
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor	84.89	44.66	51.6	68.86	61.59	55.04	38.02	56.68	37.96	6.3
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor	90.55	25.27	12.77	**74.01**	48.99	37.35	10.85	40.26	10.85	4.35
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor	**90.24**	33.94	13.43	73.23	53.51	40.2	10.67	43.53	10.64	7.83
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor	88.38	38.62	24.87	72.69	56.66	45.39	19.2	48.21	19.19	3.41
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor	89.38	33.64	20.91	73.52	53.36	42.69	17.39	45.36	17.38	9.87
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor	84.3	44.39	58.86	71.67	60.42	58.31	40.82	58.84	40.69	3.52
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor	84.69	**44.96**	**59.53**	73.54	61.73	**59.34**	41.39	**59.94**	41.22	8.21
3	AVG	ℓ_∞, StAdv, Recolor, ℓ_2	87.77	41.97	39.55	67.93	68.55	54.5	30.39	54.5	30.39	50.54
	MAX	ℓ_∞, StAdv, Recolor, ℓ_2	84.3	41.69	52.3	65.1	57.62	54.18	37.44	54.18	37.44	55.54
	Random	ℓ_∞, StAdv, Recolor, ℓ_2	86.32	35.04	47.82	68.35	65.87	54.27	30.76	54.27	30.76	12.41
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	82.27	44.21	58.13	69.08	60.7	58.03	**41.48**	58.03	**41.48**	7.9
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	82.6	43.84	57.75	68.84	60.23	57.66	41.19	57.66	41.19	19.74
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	85.11	44.71	50.32	68.39	63.29	56.68	37.23	56.68	37.23	2.37
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	85.33	43.8	50.28	68.77	63.17	56.51	36.77	56.51	36.77	5.95
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	88.49	**44.93**	18.06	65.96	67.56	49.13	15.78	49.13	15.78	1.63
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	**89.3**	42.72	11.85	60.27	**69.12**	45.99	10.71	45.99	10.71	4.07
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	88.14	41.52	26.06	61.97	68.77	49.58	24.19	49.58	24.19	2.52
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	87.8	40.78	28.34	59.47	68.32	49.23	25.92	49.23	25.92	5.89
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.56	42.19	55.55	69.95	60.69	57.1	38.24	57.1	38.24	3.4
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.1	43.32	**58.2**	**72.09**	61.96	**58.89**	39.97	**58.89**	39.97	8.28

Table 7: **Continual Robust Training on CIFAR-10 (Sequence of 4 attacks starting with ℓ_∞).** The learner initially has knowledge of ℓ_∞ attacks and over time, we are sequentially introduced to StAdv, ReColor, and ℓ_2 attacks. We report clean accuracy, accuracy on different attack types, and average and union accuracies. The threat models column represents the set of attacks known to the defender and accuracies on known attacks are highlighted with in green cells. "FT" procedures are fine-tuning approaches starting from adversarially trained to ℓ_∞ model (AT) and then sequentially fine-tuning with new attacks for 25 epochs. AVG, MAX, and Random strategies train models from scratch with all attacks for 100 epochs. The "Avg (known)" and "Union (known)" columns represent average and union accuracies on attacks known to the defender while "Avg (all)" and "Union (all)" columns represent average and union accuracies on all four attacks. Additionally, we report training times for the procedure (non-cumulative) in the "Time" column. Best performance out of both training from scratch and fine-tuning are bolded, while best performance when only comparing fine-tuning approaches is underlined.

Time Step	Procedure	Threat Models	Clean	ℓ_2	StAdv	ℓ_∞	Recolor	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	AT	ℓ_2	**90.22**	83.95	10.65	7.67	49.22	83.95	83.95	37.87	3.16	1.71
	AT + ALR	ℓ_2	89.76	**84.41**	**28.23**	**25.22**	**54.70**	**84.41**	**84.41**	**48.14**	**18.01**	2.15
1	AVG	ℓ_2, StAdv	84.56	77.68	74.32	7.57	31.33	76	73.68	47.73	7.44	3.58
	MAX	ℓ_2, StAdv	85.22	76.87	**77.63**	4.94	27.61	**77.25**	**75.57**	46.76	4.76	3.52
	Random	ℓ_2, StAdv	85.71	77.55	74.32	5.78	29.61	75.94	73.55	46.82	5.53	2.58
	FT MAX (10 ep)	ℓ_2, StAdv	83.92	77.5	69.02	10.78	35.77	73.26	68.89	48.27	10.45	0.61
	FT MAX (25 ep)	ℓ_2, StAdv	84.56	77.73	69.35	9.76	36.15	73.54	69.1	48.25	9.43	1.44
	FT Croce (10 ep)	ℓ_2, StAdv	85.07	78.62	67.52	10.57	38.34	73.07	67.31	48.76	10.29	0.4
	FT Croce (25 ep)	ℓ_2, StAdv	**86.37**	**79.67**	69.32	9.81	38.27	74.5	69.17	49.27	9.63	0.98
	FT Single (10 ep)	ℓ_2, StAdv	84.08	77.86	68.31	10.83	36.97	73.08	68.13	48.49	10.45	0.51
	FT Single (25 ep)	ℓ_2, StAdv	85.63	78.39	72.31	7.57	35.31	75.35	72.08	48.39	7.36	1.15
	FT Single + ALR (10 ep)	ℓ_2, StAdv	83.8	77.94	71.62	**20.71**	43.13	74.78	71.34	**53.35**	**20.13**	0.58
	FT Single + ALR (25 ep)	ℓ_2, StAdv	83.9	77.78	71.97	17.35	38.39	74.88	71.59	51.38	16.76	1.44
	FT Croce + ALR (10 ep)	ℓ_2, StAdv	85.04	79.54	69.99	18.68	42.93	74.76	69.89	52.78	18.09	0.51
	FT Croce + ALR (25 ep)	ℓ_2, StAdv	85.07	79.39	68	19.57	**43.67**	73.69	67.97	52.66	19.16	1.24
2	AVG	ℓ_2, StAdv, ℓ_∞	**86.62**	**84.92**	68.89	50.57	66.98	**68.13**	49.17	**67.84**	47.82	10.51
	MAX	ℓ_2, StAdv, ℓ_∞	80.36	78.09	68.38	**52.61**	67.29	66.36	**51.77**	66.59	**50.37**	11.96
	Random	ℓ_2, StAdv, ℓ_∞	84.92	83.06	68.76	49.50	66.11	67.11	48.15	66.86	46.60	4.29
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞	81.76	76.69	71.03	28.31	54.32	58.68	28.31	57.59	27.69	0.67
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞	82.04	77.86	69.02	42.83	66.9	63.24	42.37	64.15	41.86	1.71
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞	83.59	78.8	69.53	34.17	61.5	60.83	34.06	61	33.61	0.3
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞	85.22	81.02	69.58	39.92	64.79	63.51	39.59	63.83	39.03	0.73
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞	82.06	77.25	73.1	27.21	57.4	59.18	27.21	58.74	26.9	0.22
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞	82.04	77.96	70.42	41.15	66.09	63.18	40.92	63.9	40.46	0.54
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	81.38	77.89	71.8	46.68	**72.13**	65.45	46.5	67.12	46.14	0.31
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	80.92	77.43	70.78	47.16	70.6	65.12	46.96	66.49	46.62	0.79
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	83.95	79.57	69.22	37.96	59.77	62.25	37.86	61.63	36.99	0.40
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	83.11	79.24	**72.38**	36.61	60.18	62.74	36.59	62.1	36.15	1.01
3	AVG	ℓ_2, StAdv, ℓ_∞, Recolor	**87.67**	**85.66**	66.06	50.42	75.90	69.51	47.90	69.51	47.90	13.79
	MAX	ℓ_2, StAdv, ℓ_∞, Recolor	83.26	81.22	70.70	**56.94**	74.80	**70.92**	**55.31**	**70.92**	**55.31**	14.60
	Random	ℓ_2, StAdv, ℓ_∞, Recolor	86.55	84.64	66.52	47.29	74.93	68.34	45.71	68.34	45.71	9.61
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	81.99	77.78	68.28	41.83	69.91	64.45	41.4	64.45	41.4	1.31
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	82.78	79.21	**70.83**	45.15	71.39	66.64	44.76	66.64	44.76	3.6
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	84.87	80.38	66.68	36.82	68.61	63.12	36.31	63.12	36.31	0.45
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	86.32	82.11	68.79	41.27	72.41	66.15	40.69	66.15	40.69	1.2
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	86.27	81.35	54.73	23.59	70.17	57.46	22.55	57.46	22.55	0.71
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	85.1	80.48	58.17	36.38	70.62	61.41	34.45	61.41	34.45	2.03
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	85.3	81.04	49.35	40.48	74.8	61.42	35.62	61.42	35.62	0.85
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	86.78	82.8	47.82	33.12	**77.58**	60.33	29.17	60.33	29.17	2.38
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	85.3	81.3	69.35	43.13	70.85	66.16	42.62	66.16	42.62	0.53
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	85.81	81.76	67.13	45.38	73.02	66.82	44.56	66.82	44.56	1.36

Table 8: **Continual Robust Training on ImageNette (Sequence of 4 attacks starting with ℓ_2).**

Time Step	Procedure	Threat Models	Clean	ℓ_∞	StAdv	Recolor	ℓ_2	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	AT	ℓ_∞	**82.52**	56.94	**61.32**	71.62	**78.39**	56.94	56.94	**67.07**	50.32	1.70
	AT + ALR	ℓ_∞	81.52	**59.62**	60.51	**73.50**	72.69	**59.62**	**59.62**	66.58	**52.92**	2.67
1	AVG	ℓ_∞, StAdv	**85.78**	53.30	**75.69**	67.69	**81.96**	64.5	53.02	69.66	51.11	5.87
	MAX	ℓ_∞, StAdv	83.77	**58.04**	70.04	**72.76**	80.38	64.04	**56.54**	**70.31**	55.26	6.11
	Random	ℓ_∞, StAdv	83.34	52.23	73.76	67.85	79.87	62.99	51.77	68.43	50.39	2.44
	FT MAX (10 ep)	ℓ_∞, StAdv	82.27	55.03	70.52	69.78	78.27	62.78	54.17	68.4	52.66	0.62
	FT MAX (25 ep)	ℓ_∞, StAdv	82.57	55.46	71.75	69.94	78.73	63.61	54.85	68.97	53.22	1.48
	FT Croce (10 ep)	ℓ_∞, StAdv	82.29	54.62	69.2	68.87	78.32	61.91	53.35	67.75	51.9	0.37
	FT Croce (25 ep)	ℓ_∞, StAdv	83.67	54.27	71.57	69.07	79.54	62.92	53.58	68.61	52.2	0.86
	FT Single (10 ep)	ℓ_∞, StAdv	83.06	50.52	71.52	65.43	78.78	61.02	49.96	66.56	48.18	0.51
	FT Single (25 ep)	ℓ_∞, StAdv	84.00	43.59	73.68	58.14	78.85	58.64	43.46	63.57	41.27	1.16
	FT Single + ALR (10 ep)	ℓ_∞, StAdv	82.19	42.7	73.17	60.97	77.55	57.94	42.6	63.6	41.27	0.58
	FT Single + ALR (25 ep)	ℓ_∞, StAdv	81.4	51.8	69.35	66.32	76.54	60.57	51.08	66	50.04	1.47
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv	82.62	57.71	70.11	72.41	77.89	63.91	**56.54**	69.53	55.62	0.49
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv	82.37	57.55	71.64	70.93	78.14	**64.6**	**56.54**	69.57	**55.64**	1.11
2	AVG	ℓ_∞, StAdv, Recolor	**86.39**	51.80	**73.81**	**77.99**	**83.13**	**67.86**	51.31	**71.68**	51.31	11.57
	MAX	ℓ_∞, StAdv, Recolor	81.20	54.55	68.64	72.82	78.01	65.33	53.12	68.5	53.12	13.55
	Random	ℓ_∞, StAdv, Recolor	86.29	50.96	72.28	76.59	82.96	66.61	50.27	70.69	50.27	4.90
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor	82.34	55.34	71.34	72.87	78.22	66.51	54.04	69.44	54.04	1.21
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor	83.75	55.29	72.56	74.83	79.97	67.56	54.27	70.66	54.27	3.03
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor	84.28	54.01	69.96	72.56	79.72	65.51	52.13	69.06	52.13	0.5
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor	84.05	52.99	70.47	73.07	80.33	65.51	51.92	69.22	51.92	1.23
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor	83.77	53.61	65.38	73.35	79.36	64.11	50.7	67.92	50.7	0.75
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor	85.07	48.2	65.86	75.41	80.41	63.16	46.34	67.47	46.34	1.88
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor	84.94	50.8	65.71	76.33	80.15	64.28	48.31	68.25	48.31	0.88
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor	82.09	55.46	66.27	73.63	77.76	65.12	52.89	68.28	52.89	2.14
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor	82.42	**55.75**	65.83	73.3	78.06	64.96	52.94	68.24	52.94	0.61
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor	83.9	55.52	71.21	75.29	79.82	67.34	**54.32**	70.46	**54.32**	1.49
3	AVG	ℓ_∞, StAdv, Recolor, ℓ_2	**87.67**	50.42	66.06	**75.90**	**85.66**	69.51	47.90	69.51	47.90	13.79
	MAX	ℓ_∞, StAdv, Recolor, ℓ_2	83.26	56.94	**70.70**	74.80	81.22	**70.92**	**55.31**	**70.92**	**55.31**	14.60
	Random	ℓ_∞, StAdv, Recolor, ℓ_2	86.55	47.29	66.52	74.93	84.64	68.34	45.71	68.34	45.71	4.58
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	82.73	**55.08**	71.36	73.5	78.98	69.73	54.19	69.73	54.19	1.3
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	83.72	54.93	72.18	74.42	79.9	70.36	53.94	70.36	53.94	3.26
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.33	52.18	69.5	72.74	79.97	68.6	50.55	68.6	50.55	0.44
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.84	52.59	68.74	73.27	81.45	69.01	50.96	69.01	50.96	1.1
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.79	53.02	65.43	72.82	80.08	67.83	50.29	67.83	50.29	0.26
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	85.5	49.12	64.79	74.27	80.76	67.24	47.21	67.24	47.21	0.63
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	85.1	45.4	63.44	67.06	80.59	64.12	42.96	64.12	42.96	0.35
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	83.64	54.37	64.48	72.41	79.69	67.74	50.96	67.74	50.96	0.92
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	83.03	53.96	67.9	72.38	79.08	68.33	51.95	68.33	51.95	0.55
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	84.84	53.94	68.51	75.11	81.32	69.72	52.23	69.72	52.23	1.36

Table 9: **Continual Robust Training on ImageNette (Sequence of 4 attacks starting with ℓ_∞).**

Time Step	Procedure	Threat Models	Clean	ℓ_2	StAdv	ℓ_∞	Recolor	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	AT	ℓ_2	**67.75**	41.65	4.21	14.28	22.46	41.65	41.65	20.65	2.34	14.85
	Ours	ℓ_2	63.53	**42.88**	**6.16**	**19.8**	**23.36**	**42.88**	**42.88**	**23.05**	**4.37**	21.98
1	AVG	ℓ_2, StAdv	**64.48**	**35.72**	28.57	9.18	19.51	32.15	25.25	**23.24**	7.53	47.24
	MAX	ℓ_2, StAdv	61.80	33.82	31.77	7.58	17.76	**32.79**	**27.68**	22.73	6.51	47.24
	Random	ℓ_2, StAdv	62.7	31.75	**32.25**	6.50	17.62	32.00	26.28	22.03	5.79	23.56
	FT MAX (10 ep)	ℓ_2, StAdv	60.3	30.9	29.32	5.96	17.04	30.11	23.97	20.8	5.15	4
	FT MAX (25 ep)	ℓ_2, StAdv	61.14	31.69	29.93	6.21	17.84	30.81	24.84	21.42	5.3	10.71
	FT Croce (10 ep)	ℓ_2, StAdv	62.68	35.2	23.7	8.9	**19.93**	29.45	21.02	21.93	6.77	2.6
	FT Croce (25 ep)	ℓ_2, StAdv	63.39	31.38	28.95	5.7	17.98	30.16	23.87	21	5.04	5.96
	FT Single (10 ep)	ℓ_2, StAdv	60.48	18.29	30.83	2.24	13.28	24.56	16.19	16.16	1.84	2.77
	FT Single (25 ep)	ℓ_2, StAdv	60.78	18.22	31.79	1.99	13.32	25	16.34	16.33	1.6	6.91
	FT Single + ALR (10 ep)	ℓ_2, StAdv	53	29.23	24	8.45	17.18	26.61	20.12	19.71	6.71	2.77
	FT Single + ALR (25 ep)	ℓ_2, StAdv	54.44	22.03	29.1	4.46	13.6	25.56	19.21	17.3	3.78	8.73
	FT Croce + ALR (10 ep)	ℓ_2, StAdv	59.09	34.09	26.59	9.84	19.2	30.34	23.62	22.43	8.28	3.11
	FT Croce + ALR (25 ep)	ℓ_2, StAdv	58.59	34.65	26.45	**10.55**	19.65	30.55	23.67	22.82	**8.64**	7.67
2	AVG	ℓ_2, StAdv, ℓ_∞	62.09	40.34	25.13	20.88	28.84	**28.78**	16.18	28.80	14.71	69.49
	MAX	ℓ_2, StAdv, ℓ_∞	56.94	34.74	**28.46**	22.72	**30.35**	28.64	**19.66**	**29.04**	**17.55**	69.36
	Random	ℓ_2, StAdv, ℓ_∞	60.98	38.01	25.21	17.25	25.76	26.82	14.26	26.56	12.97	19.58
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞	59.9	38.94	26.55	16.95	25.72	27.48	14.78	27.04	13.07	5.2
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞	61.01	38.56	27.54	17.05	25.91	27.72	15.34	27.27	13.66	12.25
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞	62.78	39.87	20.17	14.59	23.7	24.88	10.46	24.58	9.2	2.29
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞	65.85	42.51	11.05	19.67	26.55	24.41	8.06	24.95	7.3	5.06
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞	65.62	42.95	7.55	22.03	28.06	24.18	5.57	25.15	4.98	1.49
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞	**65.93**	42.82	7.62	21.83	28.2	24.09	5.65	25.12	5.12	3.71
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	62.35	**43.56**	8.76	23.72	27.77	25.35	6.98	25.95	6.29	2.16
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	62.56	42.33	7.7	**25.06**	26.57	25.03	6.67	25.41	6.02	5.39
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞	60.67	42.06	16.66	21.18	25.89	26.63	12.59	26.45	11.21	3.05
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞	63.43	42.92	10.14	23.16	26.37	25.41	8.29	25.65	7.64	6.7
3	AVG	ℓ_2, StAdv, ℓ_∞, Recolor	65.61	40.86	22.4	20.45	37.27	30.25	14.09	30.25	14.09	101.43
	MAX	ℓ_2, StAdv, ℓ_∞, Recolor	59.12	33.89	**28.02**	**22.20**	35.00	29.78	**18.74**	29.78	**18.74**	101.43
	Random	ℓ_2, StAdv, ℓ_∞, Recolor	63.1	39.47	24.79	19.04	38.15	**30.36**	14.57	**30.36**	14.57	22.87
	FT MAX (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	61.5	39.34	26.97	17.25	33.56	29.28	14.55	29.28	14.55	8.61
	FT MAX (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	62.14	38.68	27.51	17.13	33.06	29.09	14.84	29.09	14.84	19.67
	FT Croce (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	64.82	41.09	19.78	16.55	32.26	27.42	10.57	27.42	10.57	2.42
	FT Croce (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	66.31	41.02	13.42	17.34	31.02	25.7	8.4	25.7	8.4	6.03
	FT Single (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	**69.82**	32.63	4.07	9.38	40.07	21.54	1.42	21.54	1.42	3.06
	FT Single (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	68.63	37.06	5.57	13.28	37.66	23.39	2.76	23.39	2.76	7.78
	FT Single + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	66.58	37.98	6.65	16.37	39.23	25.06	3.83	25.06	3.83	3.91
	FT Single + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	68.15	32.05	5.08	11.82	**41.5**	22.61	2.46	22.61	2.46	9.72
	FT Croce + ALR (10 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	64.11	**42.52**	10.89	21.36	34.05	27.21	8.11	27.21	8.11	3.42
	FT Croce + ALR (25 ep)	ℓ_2, StAdv, ℓ_∞, Recolor	65.33	39.4	11.41	16.84	34.15	25.45	7.35	25.45	7.35	7.41

Table 10: **Continual Robust Training on CIFAR-100 (Sequence of 4 attacks starting with ℓ_2).**

Time Step	Procedure	Threat Models	Clean	ℓ_∞	StAdv	Recolor	ℓ_2	Avg (known)	Union (known)	Avg (all)	Union (all)	Time (hrs)
0	AT	ℓ_∞	**60.95**	27.61	9.92	**33.2**	**35.6**	27.61	27.61	**26.58**	7.45	16.33
	AT + ALR	ℓ_∞	55.36	**28.01**	**11.25**	31.01	33.95	**28.01**	**28.01**	26.05	8.62	23.75
1	AVG	ℓ_∞, StAdv	**66.09**	8.65	33.19	18.42	21.58	20.92	8.18	20.46	7.47	47.59
	MAX	ℓ_∞, StAdv	57.01	22.8	29.23	30.54	**33.58**	**26.02**	**20.28**	**29.04**	**17.96**	46.97
	Random	ℓ_∞, StAdv	47.14	16.62	25.61	25.35	27.25	21.12	14.63	23.71	13.37	23.92
	FT MAX (10 ep)	ℓ_∞, StAdv	58.05	22.48	28.66	30.43	33.15	25.57	18.95	28.68	17.05	4.03
	FT MAX (25 ep)	ℓ_∞, StAdv	58.56	22.36	29.38	**30.97**	33.16	25.87	19.38	28.97	17.43	10.02
	FT Croce (10 ep)	ℓ_∞, StAdv	60.17	22.75	26.81	31.22	33.68	24.78	17.54	28.62	15.95	2.49
	FT Croce (25 ep)	ℓ_∞, StAdv	60.27	22.18	28.17	30.38	33.25	25.18	17.81	28.5	16.21	5.69
	FT Single (10 ep)	ℓ_∞, StAdv	55.87	15.43	24.29	24.34	27.54	19.86	11.9	22.9	10.95	2.77
	FT Single (25 ep)	ℓ_∞, StAdv	56.11	16.09	24.46	24.84	28.65	20.27	12.5	23.51	11.33	6.95
	FT Single + ALR (10 ep)	ℓ_∞, StAdv	56.03	3.81	35.21	18.52	17.31	19.51	3.77	18.71	3.51	75.49
	FT Single + ALR (25 ep)	ℓ_∞, StAdv	59.65	2.6	**37.96**	18.51	13.52	20.28	2.58	18.15	2.41	8.34
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv	54.86	**23.33**	27.19	30.15	31.54	25.26	18.75	28.05	16.86	2.97
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv	54.27	23.27	26.27	29.4	31.79	24.77	18.25	27.68	16.06	7.4
2	AVG	ℓ_∞, StAdv, Recolor	**68.19**	16.88	**29.53**	38.12	30.75	28.18	14.14	28.82	14.11	79.47
	MAX	ℓ_∞, StAdv, Recolor	57.96	22.38	28.92	35.24	33.9	28.85	**19.27**	30.11	**19.17**	79.5
	Random	ℓ_∞, StAdv, Recolor	47.14	16.62	25.61	25.35	27.25	21.12	14.63	23.71	13.37	23.92
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor	58.96	22.04	29	36.35	34.55	29.13	18.37	30.48	18.31	6.75
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor	59.41	21.7	29.2	35.57	33.24	28.82	18.21	29.93	18.09	16.75
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor	62.27	22.42	26.16	37.25	34.5	28.61	16.34	30.08	16.24	2.77
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor	62.09	23.24	25.6	36.91	35.89	28.58	16.63	30.41	16.52	6.46
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor	64.02	20.72	14.72	41.7	32.89	25.71	9.61	27.51	9.57	3.01
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor	67.39	13.25	6.94	45.1	27.71	21.76	3.59	23.25	3.57	7.86
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor	64.86	8.76	9.34	46.4	25.62	21.5	3.91	22.53	3.91	3.8
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor	66.87	3.74	8.44	**50.81**	19.81	21	2.03	20.7	2.03	9.89
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor	56.41	24.28	25.62	36.21	34.43	28.7	17.6	30.14	17.41	3.6
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor	56.83	22.43	25.9	38.27	33.5	**28.87**	16.98	30.03	16.76	8.28
3	AVG	ℓ_∞, StAdv, Recolor, ℓ_2	64.8	20.9	22.46	37.27	41.05	30.42	14.56	30.42	14.56	101.39
	MAX	ℓ_∞, StAdv, Recolor, ℓ_2	57.9	22.39	28.72	35.96	35.65	**30.68**	**19.15**	**30.68**	**19.15**	101.25
	Random	ℓ_∞, StAdv, Recolor, ℓ_2	63.23	19.52	21.29	**39.71**	39.95	30.12	13.6	30.12	13.6	24.88
	FT MAX (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	59.61	22.14	29.13	36.17	34.35	30.45	18.61	30.45	18.61	8.58
	FT MAX (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	59.42	22.02	**29.28**	35.96	34.45	30.43	18.64	30.43	18.64	21.36
	FT Croce (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	62.44	20.96	26.06	35.91	36.77	29.93	15.83	29.93	15.83	2.38
	FT Croce (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	62.17	21.84	26.14	36.92	36.69	30.4	16.08	30.4	16.08	5.81
	FT Single (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	63.94	23.86	13.73	37.22	41.47	29.07	9.92	29.07	9.92	1.61
	FT Single (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	**66.44**	21.17	7.72	31.83	**42.5**	25.8	5.67	25.8	5.67	4.07
	FT Single + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	60.76	22.36	10.35	31.33	41.91	26.49	7.99	26.49	7.99	2.35
	FT Single + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	62.25	20.56	7.92	30.69	41.42	25.15	6.25	25.15	6.25	6.35
	FT Croce + ALR (10 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	57.56	24.64	22.52	35.77	37.55	30.12	15.96	30.12	15.96	3.36
	FT Croce + ALR (25 ep)	ℓ_∞, StAdv, Recolor, ℓ_2	58.14	**24.85**	18.69	36.75	39.16	29.86	13.87	29.86	13.87	7.64

Table 11: **Continual Robust Training on CIFAR-100 (Sequence of 4 attacks starting with ℓ_∞).**

noise variance from regularization strength. We report results in Table 12. Overall, we find that σ has an effect similar to the effect of increasing λ where higher values of σ leads to higher average and union robust accuracies at the cost of lower clean accuracy and accuracy on the initial attack.

Noise type	σ	Clean	ℓ_2	ℓ_∞	StAdv	Recolor	Avg	Union
Uniform	0.5	90.40	70.21	31.91	1.31	39.83	35.81	0.86
Uniform	1	90.76	69.89	32.58	1.49	40.14	36.02	1.15
Uniform	2	85.28	63.65	50.73	10.62	60.10	46.28	8.40
Gaussian	0.05	90.04	69.62	31.61	7.25	43.84	38.08	6.64
Gaussian	0.1	88.53	68.54	32.04	14.5	51.73	41.70	12.63
Gaussian	0.2	87.00	64.88	27.46	31.82	63.59	46.94	18.7

Table 12: **Impact of σ on regularization based on random noise in initial training.** We maintain regularization strength $\lambda = 5$ and perform initial training on CIFAR-10 with ℓ_2 attacks. We report the clean accuracy, accuracy on ℓ_2, ℓ_∞, StAdv, and Recolor attacks, and the average and union accuracies on the set.

I.2. Comparison to TRADES

In this section, we compare ALR regularizer to TRADES regularizer (Zhang et al., 2019). TRADES is designed for improving clean accuracy tradeoff while ALR is designed for improving generalization across (seen and unforeseen) attacks. Since TRADES regularizer also maximizes a distance (KL instead of L2) in the logit space, we expect it can also improve generalization across attacks as well and provide results below. Similar to experiments with ALR, we add the regularizer on top of PGD L2 and Linf adversarial training. We present results in Table 13 with regularization strength in parentheses next to each regularization method. Notably, increasing TRADES strength in Linf training trades off Linf performance, whereas ALR does not.

Threat model	Regularizer	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Union
ℓ_2	None	91.17	69.7	28.41	2.08	44.94	1.24
ℓ_2	Trades (1)	90.43	70.08	31.33	0.89	38.51	0.6
ℓ_2	Trades (3)	88.93	70.05	33.81	9.04	58.25	6.74
ℓ_2	Trades (6)	88.76	69.69	33.00	7.04	56.82	5.51
ℓ_2	ALR (1)	89.43	69.84	34.00	48.23	65.46	31.27
ℓ_∞	None	85.93	59.48	51.44	14.87	62.48	11.9
ℓ_∞	Trades (1)	85.39	59.33	49.23	14.11	64.45	11.45
ℓ_∞	Trades (3)	83.97	58.54	47.00	20.51	69.33	16.34
ℓ_∞	Trades (6)	85.72	56.44	41.70	23.17	70.23	17.83
ℓ_∞	ALR (0.5)	83.18	58.15	51.49	34.78	58.15	29.87

Table 13: **Comparison to TRADES.** We compare robustness measured across different threat models when initial training on ℓ_2 and ℓ_∞ with either TRADES and or ALR regularizer.

I.3. Performance across different threat models

In this section, we perform initial training with models using different initial attacks including attacks in the UAR benchmark (Kaufmann et al., 2019) and evaluate the performance across attack types for training with single-step variation regularization, single-step adversarial ℓ_2 regularization, uniform regularization, and gaussian regularization.

We present ablation results for CIFAR-10 (Table 15 for variation regularization, Table 14 for adversarial ℓ_2 regularization, Table 16 for uniform regularization, and Table 17 for Gaussian regularization) and ImageNette (Table 19 for variation regularization, Table 18 for adversarial ℓ_2 regularization, Table 20 for uniform regularization, and Table 21 for Gaussian regularization). Overall, we find that across different starting attacks and unseen test attacks, regularization generally improves performance on unseen attacks, leading to increases in average and union accuracy across all attacks with regularization. We find that in many cases (especially using random noise types) using regularization trades off clean accuracy. Additionally, some threat models such as Snow are generally more difficult to gain improvement on via

regularization; for many starting models, using regularization decreases accuracy on Snow attack.

I.4. Impact of starting and new attack pairs

In order to see how much our results depend on attack choice, we experiment with starting with a model initially trained with a starting attack and then fine-tuned for robustness to a new attack for different starting and new attack pairs on Imagenette. In this section, we ask the question: does regularization in initial training generally lead to better starting points for fine-tuning? In the following experiments, we use adversarial training as the base initial training procedure and Croce & Hein (2022)'s fine-tuning approach as the base fine-tuning procedure. We consider these approaches with and without regularization.

We compare models initially trained with regularization (and fine-tuned without regularization) to models initially trained without regularization (and fine-tuned without regularization). We present the differences in average accuracy across the 2 attacks, union accuracy across the 2 attacks, accuracy on the starting attack, accuracy on the new attack, and clean accuracy between the 2 settings for adversarial ℓ_2 regularization (with $\lambda = 0.5$) in Figure 6, for variation regularization in Figure 7 (with $\lambda = 0.2$), for uniform regularization (with $\sigma = 2$ and $\lambda = 1$) in Figure 8, and for gaussian regularization (with $\sigma = 0.2$ and $\lambda = 0.5$) in Figure 9. In these figures, we highlight gains in accuracy larger than 1% in green and drops in accuracy larger than 1% in red.

Regularization in initial training generally improves performance. Across all figures, we can see that for most pairs of attacks, regularization leads to improvements on average accuracy, union accuracy, accuracy on the initial attack, accuracy on the new attack. We find that this improvement is more consistent across attack types when using adversarial versions of regularization such as adversarial ℓ_2 regularization or variation regularization in comparison to random noise based regularizations. This improvement in performance may be due to the fact that regularization improves unforeseen robustness, causing the initial accuracy on the new attack to generally be higher, and thus a better starting point for fine-tuning the model for robustness against new attacks.

Uniform regularization in initial training can improve clean accuracy for certain starting attack types. From Figure 8e, we observe that using uniform regularization in initial training can lead to increases in clean accuracy after fine-tuning for several initial attack types: StAdv, ReColor, Pixel, Elastic, Wood, and Kaleidoscope attacks. In comparison, Figure 6e, demonstrates that using adversarial ℓ_2 regularization does not improve clean accuracy for as many threat models as uniform regularization; for adversarial ℓ_2 regularization, the most improvements in clean accuracy are when the initial attack is Elastic attack or when the new attack is ℓ_∞ attack. Adversarial ℓ_2 regularization generally maintains clean accuracy for most attacks, but leads a drop in clean accuracy when the starting attack type is StAdv attack. We find that similarly, variation regularization also maintains clean accuracy. Gaussian regularization on the other hand either maintains or exhibits a tradeoff with clean accuracy.

J. Fine-tuning Ablations

J.1. Impact of starting and new attack pairs

Similar to Appendix I.4, we ablate over starting and new attack pairs in finetuning. In this section, we address the question: does regularization in fine-tuning generally lead to more robust models? We follow the same setup as in Appendix I.4 but we compare models fine-tuned with regularization (with no regularization in pretraining) to models fine-tuned without regularization (with no regularization in pretraining). We present the differences in average accuracy across the 2 attacks, union accuracy across the 2 attacks, accuracy on the starting attack, accuracy on the new attack, and clean accuracy between the 2 settings for adversarial ℓ_2 regularization (with $\lambda = 0.5$) in Figure 10 and for uniform regularization (with $\sigma = 2$ and $\lambda = 1$) in Figure 13. In these figures, we highlight gains in accuracy larger than 1% in green and drops in accuracy larger than 1% in red.

Adversarial ℓ_2 regularization in fine-tuning generally improves performance but trades off clean accuracy. From Figure 10, we can see that for many pairs of initial and new attack, regularization leads to improvements in union accuracy, average accuracy, and new attack accuracy. However, this comes at a clear tradeoff with clean accuracy. For accuracy on the initial attack, it is difficult to see clear trends; depending on threat models there can be gains in robustness or drops in robustness. For example, when the new attack is ℓ_∞, we find that the initial attack accuracy generally drops. We find that variation regularization can also lead to gains in performance, but these gains are much less consistent than compared to adversarial ℓ_2 regularization.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**91.08**	70.02	29.38	0.79	33.69	66.93	24.59	14.99	64.22	45.13	70.85	**80.3**	30.08	44.25	0.1
ℓ_2	0.1	90.4	69.7	31.78	2.27	38.16	62.99	25.77	16.91	65.35	45.94	71.05	79.72	30.17	44.98	0.74
ℓ_2	0.2	89.49	**70.49**	33.43	4.29	42.8	**68.03**	26.85	18.79	66.04	47.27	72.21	79.65	**33.38**	46.94	1.78
ℓ_2	0.5	89.57	70.29	**34.16**	**17.44**	**51.04**	65.63	**28.71**	**22.5**	**66.76**	**48.8**	**73.24**	79.66	28.83	**48.92**	**5.94**
ℓ_∞	0	**85.53**	**59.36**	50.98	6.34	56.27	**68.94**	36.79	20.57	**54.02**	51	**64.24**	**75.94**	**39.44**	48.66	1.31
ℓ_∞	0.1	85.06	58.77	51.44	7.43	55.59	68.33	37.09	20.11	53.89	51.84	64.38	74.96	42.43	48.86	1.95
ℓ_∞	0.2	85.23	58.08	**51.49**	8.96	56.32	68.3	37.11	21.48	52.86	51.61	63.72	75.49	40.22	48.8	2.44
ℓ_∞	0.5	83.18	58.21	51.47	**19.5**	**61.02**	68.75	**37.94**	**22.78**	53.89	**49.82**	63.47	73.57	39.88	**50.02**	**5.52**
StAdv	0	**87.12**	5.48	0.07	56.22	5.69	17.62	57.8	**5.93**	11.09	**76.02**	**77.47**	**54.04**	43.4	**34.24**	0.05
StAdv	0.1	86.95	4.63	0.08	56.16	4.44	20.44	57.25	4.93	9.27	75.2	76.66	52.77	40.68	33.54	0.06
StAdv	0.2	81.39	**5.99**	**0.1**	54.98	**8.16**	15.97	48.37	5.81	11.07	68.05	72.03	48.7	**45.22**	32.04	**0.09**
StAdv	0.5	85.05	5.35	0.07	**56.49**	5.09	**23.63**	**58.09**	5.51	**11.29**	73.77	75.86	51.32	42.85	34.11	0.06
ReColor	0	93.61	37.17	7.03	0.01	67.48	55.53	37.14	8.27	45.36	35.55	60.92	77.2	**32.28**	38.66	0
ReColor	0.1	93.79	35.12	6.7	0	67.12	51.54	37.64	8.69	43.64	36.17	**63.33**	76	28.53	37.87	0
ReColor	0.2	**93.84**	**37.7**	**7.87**	0.01	68.67	**55.97**	38.3	9.81	46.85	**38.54**	60.01	**77.73**	31.13	39.38	0.01
ReColor	0.5	94.57	32.67	5.83	**32.12**	**73.79**	52.74	**38.66**	**20.19**	**50.07**	35.85	61.98	75.72	24.93	**42.05**	**2.27**
Gabor	0	**94.08**	0.3	0.01	0.01	4.43	**92.39**	16.96	8.96	2.08	2.31	17.99	41.61	**11.87**	16.58	0
Gabor	0.1	93.33	0.75	0.02	0.28	29.93	91.15	17.97	16.13	11.13	8.11	20.77	46.33	10.78	21.11	0
Gabor	0.2	93.38	1.17	0.01	9.4	54.87	91.15	**25.47**	**33.47**	26.37	**19.74**	21.57	51.05	9.43	**28.64**	0
Gabor	0.5	93.27	**1.57**	**0.03**	**10.16**	**56.04**	91.52	24.9	30.66	**26.36**	15.33	**24.18**	**51.55**	11.15	28.62	**0.01**
Snow	0	**95.89**	0.05	0	0.01	2.63	**30.13**	**92.02**	**7.23**	0.9	15.17	31.49	**47.98**	22.39	20.83	0
Snow	0.1	94.68	0.17	0	0.12	5.99	23.92	89.23	3.96	1.17	26.82	45.03	42.74	23.93	21.92	0
Snow	0.2	94.51	0.21	0	6.28	**20.53**	29.14	89.7	7.12	**6.48**	**40.55**	51.42	45.23	**25.96**	**26.89**	0
Snow	0.5	88.84	**0.53**	0	**9.13**	15.16	19.39	83.83	3.6	6.13	39.54	**57.62**	33.58	20.73	24.1	0
Pixel	0	**94.76**	0.07	0	0.01	8.87	**57.65**	36.6	88.35	1.77	14.52	38.18	67.5	16.21	27.48	0
Pixel	0.1	94.47	**0.27**	0	1.86	31.03	52.29	41.44	88.01	7.59	26.96	41.24	**68.23**	19.99	31.58	0
Pixel	0.2	94.01	**0.27**	0	**5.57**	**34.43**	51.54	**43.31**	**88.53**	**8.9**	**28.75**	43.53	66.73	18.36	**32.49**	0
Pixel	0.5	92.34	0.06	0	5.38	19.85	44.07	38.11	87.21	4.85	27.22	**49.64**	58.39	**20.92**	29.64	0
JPEG	0	**90.26**	56.48	21.5	0.52	34.74	68.59	21.12	10.57	73.46	40	74.3	**78.35**	**28.02**	42.3	0.09
JPEG	0.1	89.41	58.2	24.43	1.24	37.73	**73.19**	22	12.59	74.05	40.57	75.02	77.88	27.61	43.71	0.41
JPEG	0.2	88.56	58.55	26.21	3.19	41.12	71.49	22.1	14.65	74.23	40.7	75.43	78.17	24.93	44.23	1.08
JPEG	0.5	87.33	**60.43**	**29.14**	**11.66**	**46.74**	72.68	**24.34**	**17.81**	**74.37**	**43.52**	**75.44**	77.08	25.76	**46.58**	**3.39**
Elastic	0	**94.06**	1.32	0.02	7.5	7.92	25.41	53.68	9.16	11.2	79.47	72.94	50.24	33.1	29.33	0.01
Elastic	0.1	93.49	1.68	**0.03**	**51.42**	**41.84**	**28.14**	**56.78**	**14.2**	**26.91**	80.19	74.93	53.78	29.16	**38.25**	0.02
Elastic	0.2	93.32	1.87	0.01	17.64	11.84	25.67	55.38	5.61	9.2	**80.66**	77.2	51.52	**36.14**	31.06	0.01
Elastic	0.5	92.62	**2.64**	0.1	40.11	28.38	27.2	51.69	10.24	20.46	80.3	**77.69**	**55.69**	34.25	35.73	**0.08**
Wood	0	**93.57**	0.03	0	0.4	1.27	**18.47**	39.44	3.43	0.37	33.68	**93.04**	28.04	14.11	19.36	0
Wood	0.1	92.79	**0.04**	0	3.84	4.88	16.77	42.61	3.32	1.21	37.35	92.3	29.72	**16.04**	20.67	0
Wood	0.2	92.68	0.02	0	10.98	11.85	15.74	44.47	5.55	3.76	38.77	92.25	32.48	14.83	22.56	0
Wood	0.5	92.03	**0.04**	0	**25.21**	**21.39**	17.58	**47.09**	**7.52**	**6.2**	**43.25**	91.36	**36.32**	15.18	**25.93**	0
Glitch	0	**93.26**	0.02	0	0	11.49	49.03	24.44	12.47	3.14	10.89	31.99	**90.77**	16.61	20.9	0
Glitch	0.1	92.06	0.07	0	0	8.05	45.58	20.58	5.59	1.16	9.27	32.45	87.47	16.96	18.93	0
Glitch	0.2	90.33	**1.09**	**0.09**	0.01	17.43	54.43	20.15	8.3	7.71	21.52	47.55	83.74	19.34	23.45	0
Glitch	0.5	92.03	0.36	0.02	**16.62**	**45.09**	**54.52**	**25.11**	**18.73**	**21.68**	**27.67**	**47.92**	87.59	**19.87**	**30.43**	0
Kaleid-oscope	0	96.03	0	0	0	0.8	39.49	40.94	5.75	0.02	2.4	43.08	33.71	91.97	21.51	0
Kaleid-oscope	0.1	**96.22**	0.03	0	2.15	33.97	37.41	35.7	16.06	5.21	13.03	50.03	45.1	**93.13**	27.65	0
Kaleid-oscope	0.2	96.14	0.07	0	7.48	43.57	39.96	38.95	18.03	8.43	16.61	**50.22**	48.81	92.94	30.42	0
Kaleid-oscope	0.5	95.71	**0.09**	0	**27.92**	**63.33**	**41.84**	**43.26**	**25.65**	**16.73**	**23.98**	48.61	**52.28**	92.43	**36.34**	0

Table 14: **Intial Training Ablations- adversarial ℓ_2 regularization on CIFAR-10.** Accuracy of initially trained models on CIFAR-10 trained using different attacks as indicated in "Train Attack" column measured across different attacks. ℓ_2 regularization computed using single step optimization is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**91.08**	70.02	29.38	0.79	33.69	66.93	24.59	14.99	64.22	45.13	70.85	**80.3**	30.08	44.25	0.1
ℓ_2	0.05	90.15	69.61	32.39	1.88	40.16	65.99	27.04	19.33	64.8	47.01	72.13	79.82	30.18	45.86	0.72
ℓ_2	0.1	89.24	70.36	33.37	4.27	42.96	64.75	**29.09**	19.85	66.28	**49.46**	72.01	79.6	**32.48**	47.04	1.92
ℓ_2	0.2	89.99	**70.38**	**34.56**	**13.41**	**48.99**	**67.64**	**29.09**	**22.57**	**66.64**	48.38	**73.31**	80.07	32.33	**48.94**	**5.4**
ℓ_∞	0	**85.53**	**59.36**	50.98	6.34	56.27	**68.94**	36.79	20.57	54.02	51	**64.24**	**75.94**	39.44	48.66	1.31
ℓ_∞	0.05	84.57	58.68	51.28	7.82	55.74	65.31	38.58	21.17	54.41	**52.59**	63.87	74.38	39.5	48.61	1.92
ℓ_∞	0.1	84.98	57.76	51.52	11.45	57.69	67.39	**39.27**	22.22	53.62	51.09	60.19	74.73	41.48	49.03	3.45
ℓ_∞	0.2	82.58	58.36	**51.53**	**18.98**	**62.12**	67.18	39.22	**23.62**	**54.73**	52	63.35	71.72	**43.18**	**50.5**	**5.08**
StAdv	0	**87.12**	5.48	0.07	56.22	5.69	17.62	**57.8**	5.93	11.09	**76.02**	**77.47**	54.04	**43.4**	34.24	0.05
StAdv	0.05	75.6	3.52	0.02	69.34	23.98	15.36	37.22	3.69	9.73	55.75	66.64	37.79	26.08	29.09	0.02
StAdv	0.1	81.12	8.78	**0.24**	**69.82**	15.75	27.34	48.4	5.1	**29.68**	66.67	77.1	53.77	34.7	36.45	**0.17**
StAdv	0.2	84.4	**10.93**	0.19	69.1	**28.03**	**33.19**	47.19	**6.06**	28.29	66.46	76.31	**58.5**	40.51	**38.73**	0.14
ReColor	0	93.61	**37.17**	7.03	0.01	67.48	55.53	37.14	8.27	45.36	35.55	60.92	77.2	**32.28**	38.66	0
ReColor	0.05	**93.68**	35.65	**7.10**	0.07	69.73	51.67	36.85	9.26	44.65	**38.42**	60.68	**77.36**	31.85	38.61	0.02
ReColor	0.1	93.63	33.32	7.00	19.46	77.93	**57.73**	36.18	17.35	47.63	34.61	**61.9**	76.26	27.3	41.39	**2.22**
ReColor	0.2	92.67	29.88	5.83	**30.85**	**83.95**	55.41	**37.24**	**18.71**	**48.31**	34.9	60.33	74.58	27.93	**42.33**	2.07
Gabor	0	**94.08**	0.3	0.01	0.01	4.43	**92.39**	16.96	8.96	2.08	2.31	17.99	41.61	**11.87**	16.58	0
Gabor	0.05	93.84	0.41	0.01	0.17	20.82	91.9	17.71	14.23	7.61	6.36	16.61	45.52	10.71	19.34	0
Gabor	0.1	93.69	1.14	**0.03**	4.47	45.21	91.2	22.26	27.21	22.28	19.2	19.73	47.75	9.99	25.87	0
Gabor	0.2	93.61	**1.35**	0.02	**13.59**	**57.17**	90.85	**29.97**	**33.81**	**31.63**	**25.28**	**22.52**	**52.4**	9.08	**30.64**	0
Snow	0	**95.89**	0.05	0	0.01	2.63	**30.13**	**92.02**	7.23	0.9	15.17	31.49	47.98	22.39	20.83	0
Snow	0.05	89.84	0.12	0	0.01	0.81	24.79	82.66	0.95	0.14	14.89	28.76	35.46	21.3	17.49	0
Snow	0.1	88.07	**0.28**	0	0.18	1.39	18.33	83.28	1.12	0.88	29.73	**56.23**	27.04	19.81	19.86	0
Snow	0.2	94.56	0.21	0	**29.86**	**42.55**	23.26	90.96	**14.4**	**15.56**	**49.51**	50.92	**50.35**	**26.23**	**32.82**	0
Pixel	0	**94.76**	**0.07**	0	0.01	8.87	57.65	36.6	88.35	1.77	14.52	38.18	**67.5**	16.21	27.48	0
Pixel	0.05	93.81	0.01	0	0.03	12.65	**59.31**	36.8	88.88	3.83	14.19	40.33	62.99	16.9	27.99	0
Pixel	0.1	93.54	0.02	0	0.22	15.32	54.00	38.37	**89.45**	3.9	14.35	38.81	63.48	17.75	27.97	0
Pixel	0.2	93.06	0.06	**0.01**	**3.59**	**22.18**	50.55	**40.96**	89.39	**9.85**	**16.49**	**40.79**	61.44	**18.25**	**29.46**	0
JPEG	0	**90.26**	56.48	21.5	0.52	34.74	68.59	21.12	10.57	73.46	40	74.3	78.35	**28.02**	42.3	0.09
JPEG	0.05	89.52	57.89	23.81	1.19	37.2	71.84	21.88	13.16	73.9	38.82	74.05	**78.61**	27.97	43.36	0.46
JPEG	0.1	89.09	58.18	25.89	4.28	40.45	**74.3**	20.85	14.35	**74.42**	40.82	75.08	78.07	25.35	44.34	1.42
JPEG	0.2	87.98	**58.94**	**26.85**	**10.19**	**44.26**	70.96	**22.18**	**16.37**	74.17	**41.13**	**76.03**	77.65	24.9	**45.3**	**2.74**
Elastic	0	94.06	1.32	**0.02**	**7.5**	7.92	**25.41**	**53.68**	9.16	11.2	79.47	72.94	**50.24**	**33.1**	**29.33**	**0.01**
Elastic	0.05	93.88	**1.49**	0.01	7.27	**8.18**	23.48	51.12	**9.32**	**11.5**	79.77	73.88	50.2	30.32	28.88	**0.01**
Elastic	0.1	**94.13**	1.2	0	7.25	7.87	20.68	53.62	8.52	10.93	**79.93**	**74.77**	49.44	30.54	28.73	0
Elastic	0.2	93.79	1.23	0.01	6.49	6.77	21.8	53.29	7.64	10.75	79.76	73.77	49.72	32.75	28.67	0
Wood	0	93.57	0.03	0	0.4	1.27	18.47	39.44	3.43	0.37	33.68	**93.04**	28.04	14.11	19.36	0
Wood	0.05	92.78	0.01	0	0.65	1.65	16.47	41.6	2.41	0.51	35.08	92.36	27.79	**17.15**	19.64	0
Wood	0.1	92.59	0.02	0	3.3	**5.25**	18.41	43.33	4.07	1.67	37.47	92.02	29.95	14.92	20.87	0
Wood	0.2	91.92	**0.03**	0	**8.21**	**8.76**	**15.98**	42.64	**5.12**	**2.58**	**38.15**	91.5	**32.01**	14.15	**21.59**	0
Glitch	0	93.26	0.02	0	0	11.49	49.03	24.44	12.47	3.14	10.89	31.99	90.77	**16.61**	20.9	0
Glitch	0.05	92.38	0.01	0	0.03	16.86	48.73	23.12	9.96	5.16	14.01	31.82	90.8	15.63	21.34	0
Glitch	0.1	92.14	0.02	0	0.23	24.38	48.71	22.89	10.69	5.38	17.19	**32.08**	**91.19**	14.59	22.28	0
Glitch	0.2	92.62	**0.04**	0	**5.11**	**39.62**	**56.19**	**26.18**	**18.76**	**14.5**	**26.28**	31.86	90.95	16.25	**27.15**	0
Kaleid-oscope	0	96.03	0	0	0	0.8	39.49	**40.94**	5.75	0.02	2.4	43.08	33.71	91.97	**21.51**	0
Kaleid-oscope	0.05	**96.31**	0	0	0	**1.59**	39	35.73	6.29	0.02	1.88	40.31	36.53	**93.07**	21.2	0
Kaleid-oscope	0.1	96.22	0	0	0	1.4	**39.98**	33.78	**6.75**	0.01	1.9	**43.17**	35.1	92.83	21.24	0
Kaleid-oscope	0.2	96.01	0	0	0	1.33	34.92	35.41	6.01	**0.1**	**2.61**	38.35	**36.8**	92.7	20.69	0

Table 15: **Intial Training Ablations- variation regularization on CIFAR-10.** Accuracy of initially trained models on CIFAR-10 trained using different attacks as indicated in "Train Attack" column measured across different attacks. Variation regularization computed using single step optimization is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**91.08**	70.02	29.38	0.79	33.69	66.93	24.59	14.99	64.22	45.13	70.85	**80.3**	30.08	44.25	0.1
ℓ_2	1	90.4	**70.22**	**32.73**	6.17	41.96	**69.81**	25.46	18.13	65.98	45.71	**72.72**	79.51	30.13	46.54	2.48
ℓ_2	2	89.45	69.44	31.97	12.15	51.04	69.03	25.71	19.54	**66.11**	**47.5**	71.44	79.25	30.8	47.83	3.21
ℓ_2	5	88.34	66.66	27.41	**26.22**	**60.22**	69.16	**26.67**	**22.57**	64.08	46.83	71.14	77.6	**31.36**	**49.16**	**6.23**
ℓ_∞	0	**85.53**	59.36	**50.98**	6.34	56.27	**68.94**	36.79	20.57	54.02	51	64.24	75.94	**39.44**	48.66	1.31
ℓ_∞	1	85.28	**63.66**	50.72	10.51	60.1	66.63	36.67	21.66	60.69	51.72	**65.61**	**76.04**	38.41	50.2	3.01
ℓ_∞	2	81.97	63.5	48.24	17.33	64.72	65.73	**38.7**	24.49	**61.25**	**51.96**	63.63	72.82	39.16	50.96	4.97
ℓ_∞	5	78.04	60.28	40.59	**42.25**	**70**	67.06	33.4	**26.57**	60.07	49.21	64.61	67.08	38.43	**51.63**	**8.36**
StAdv	0	**87.12**	5.48	0.07	56.22	5.69	17.62	**57.8**	5.93	11.09	**76.02**	**77.47**	54.04	**43.4**	34.24	0.05
StAdv	1	81.36	24.44	1.29	**73.69**	53.63	40.94	39.39	10.47	39.05	65.18	72.43	62.43	36.21	43.26	0.88
StAdv	2	82.8	**38.42**	**3.9**	71.03	**55.3**	**51.03**	39.71	**12.12**	**50.4**	63.57	74.24	**68.98**	38.32	**47.25**	**2.11**
ReColor	0	**93.61**	37.17	7.03	0.01	67.48	55.53	**37.14**	8.27	45.36	35.55	60.92	77.2	32.28	38.66	0
ReColor	1	92.96	51.36	13.58	5.96	73.47	63.31	31.94	8.73	**59.67**	42.6	65.49	**78.37**	34.22	44.06	1.43
ReColor	2	92.34	53.34	14.88	17.02	78.21	63.32	34.31	13.28	58.69	44.23	65.48	77.02	**37.79**	46.46	3.7
ReColor	5	86.49	**54.92**	**16.5**	**44.41**	**78.3**	**68.38**	32.22	**27.48**	56.59	**45.33**	**65.82**	73.44	27.4	**49.23**	**6.81**
Gabor	0	**94.08**	0.3	0.01	0.01	4.43	**92.39**	16.96	8.96	2.08	2.31	17.99	41.61	11.87	16.58	0
Gabor	1	91.84	**24.27**	**0.69**	**10.02**	**41.71**	87.58	**23.11**	**13.99**	**32.19**	**27.95**	**47.51**	**62.62**	**15.27**	**32.24**	**0.24**
Gabor	2	90.58	9.65	0.06	0.17	17.39	87.02	16.34	5.68	9.84	15.66	42.22	52.81	14.41	22.6	0
Snow	0	**95.89**	0.05	0	0.01	2.63	30.13	**92.02**	7.23	0.9	15.17	31.49	47.98	22.39	20.83	0
Snow	0.5	90.18	26.34	0.63	4.74	22.47	**70.47**	79.6	7.67	25.69	39.56	46.21	**69.51**	14.55	33.95	0.13
Snow	1	86.82	25.02	0.93	**14.74**	**40.4**	65.19	74.04	**8.76**	31.82	**41.78**	**50.21**	67.76	22.3	**36.91**	0.36
Snow	2	75.97	**38.33**	**8.48**	3.21	24.12	57.66	66.64	8.1	**34.75**	39.83	47.16	60.87	**25.36**	34.54	**0.85**
Pixel	0	**94.76**	0.07	0	0.01	8.87	57.65	**36.6**	**88.35**	1.77	14.52	38.18	67.5	16.21	27.48	0
Pixel	1	88.35	11.76	0.16	0.38	23.23	**63.75**	28.4	71.23	7.2	31.88	47.73	**71.87**	21.77	31.61	0.02
Pixel	2	79.87	**22.8**	**3.38**	**1.75**	**34.1**	59	24	60.74	**13.63**	**33.78**	**52.8**	66.58	**31.06**	**33.63**	**0.22**
JPEG	0	**90.26**	56.48	21.5	0.52	34.74	68.59	21.12	10.57	73.46	40	74.3	**78.35**	28.02	42.3	0.09
JPEG	1	89.29	60.05	26.16	4.69	42.29	**74.59**	22.64	13.53	**74.29**	43.93	**74.64**	77.74	**29.07**	45.3	1.33
JPEG	2	88.49	**61.79**	**27.73**	**12.37**	**47.32**	71.46	**25.63**	**17.62**	73.98	**45.12**	72.8	78.03	27.9	**46.81**	**3.87**
Elastic	0	**94.06**	1.32	0.02	7.5	7.92	25.41	**53.68**	9.16	11.2	**79.47**	72.94	50.24	33.1	29.33	0.01
Elastic	1	91.85	17.14	0.54	5.55	14.6	40.8	35.05	5.55	31.41	66.9	**75.26**	**65.89**	30.44	32.43	0.17
Elastic	2	67.79	**43.66**	**17.81**	**23**	**31.04**	**56.14**	28.89	**15.31**	**47.71**	51.31	57.9	59.33	**34.56**	**38.89**	**4.6**
Wood	0	**93.57**	0.03	0	0.4	1.27	18.47	**39.44**	3.43	0.37	33.68	**93.04**	28.04	14.11	19.36	0
Wood	1	89.61	**30.94**	**3.52**	**17.14**	**30.98**	26.2	4.9	**53.36**	**45.05**	**56.82**	82.92	**66.69**	**25.1**	**36.97**	**0.69**
Wood	2	85.72	**30.94**	**3.52**	**17.14**	**30.98**	**53.36**	26.2	4.9	**45.05**	**56.82**	82.92	**66.69**	**25.1**	**36.97**	**0.69**
Glitch	0	**93.26**	0.02	0	0	11.49	49.03	**24.44**	12.47	3.14	10.89	31.99	**90.77**	16.61	20.9	0
Glitch	1	90.16	34.66	2.61	0.28	22.95	**64.19**	22.01	12.38	28.75	39.22	**60.2**	83.26	23.6	32.84	0.01
Glitch	2	79.67	**44.33**	**9.64**	**1.01**	**27.31**	60.07	17.26	**15.28**	**38.23**	**42.31**	58.64	72.43	**30.1**	**34.72**	**0.36**
Kaleid-oscope	0	**96.03**	0	0	0	0.8	39.49	**40.94**	5.75	0.02	2.4	43.08	33.71	**91.97**	21.51	0
Kaleid-oscope	0.1	93.16	5.39	0.02	0.02	4.89	**55.54**	35.63	5.32	3.25	17.43	**52.59**	60.06	86.77	27.24	0
Kaleid-oscope	0.5	88.9	**16.02**	0.22	0.17	9.98	54.63	29.68	6.35	9.75	**29.38**	49.24	**64.47**	78.4	**29.02**	0
Kaleid-oscope	1	68.25	14.05	**0.51**	**5.33**	**42**	48.53	14.55	**10.46**	**13.6**	21.92	32.66	45.02	62.09	25.89	**0.09**

Table 16: **Intial Training Ablations- Uniform regularization on CIFAR-10.** Accuracy of initially trained models on CIFAR-10 trained using different attacks as indicated in "Train Attack" column measured across different attacks. Uniform regularization (with $\sigma = 2$) is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**91.08**	70.02	29.38	0.79	33.69	66.93	24.59	14.99	64.22	45.13	70.85	**80.3**	30.08	44.25	0.1
ℓ_2	0.1	90.24	**70.25**	31.05	3.01	40.33	70.49	24.6	16.78	**65.57**	44.87	73.93	79.68	29.94	45.86	1.21
ℓ_2	0.2	90.07	69.57	31.8	9.58	46.6	72.49	25.22	18.95	65.22	46.79	**76.23**	79.36	28.74	47.55	3
ℓ_2	0.5	86.89	68.19	**32.02**	**16.54**	**58.32**	**74.85**	**25.69**	**21.26**	65.32	**46.82**	74.08	76.99	**31.93**	**49.33**	**4.18**
ℓ_∞	0	85.53	59.36	50.98	6.34	56.27	**68.94**	36.79	20.57	54.02	51	64.24	**75.94**	39.44	48.66	1.31
ℓ_∞	0.1	**86.18**	60.45	**51.52**	7.07	57.5	68.54	38.96	21.24	56.43	50.39	64.66	75.78	38.74	49.27	1.8
ℓ_∞	0.2	85.19	**61.63**	50.12	17.67	67.92	69.6	**40.02**	23.22	**57.72**	**52.76**	**66.28**	75	**40.86**	51.9	3.93
ℓ_∞	0.5	80.65	59.74	46.12	**34.57**	**70.49**	68.33	35.8	**26.04**	57.28	51.98	65.46	70.73	38.21	**52.06**	**6.28**
StAdv	0	**87.12**	5.48	0.07	56.22	5.69	17.62	**57.8**	5.93	11.09	**76.02**	**77.47**	54.04	**43.4**	34.24	0.05
StAdv	0.1	72.19	1.16	0.02	62.36	42.38	44.84	35.9	10.54	16.17	52.73	65.04	50.52	34.58	34.69	0.02
StAdv	0.2	79.63	11.11	0.42	**72.58**	57.96	43.27	40.28	11.09	25.26	60.64	70.43	57.7	36.15	40.57	0.34
StAdv	0.5	76.35	**30.61**	**4.69**	66.84	**61.11**	**50.18**	36.59	**16.76**	**39.22**	57	67.71	**62.31**	34.1	**43.92**	**3.13**
ReColor	0	**93.61**	37.17	7.03	0.01	67.48	55.53	**37.14**	8.27	45.36	35.55	60.92	**77.2**	**32.28**	38.66	0
ReColor	0.1	93.51	36.05	8.59	24.25	79.15	65.07	36.75	17.11	49.12	35.86	65.18	76.15	31.97	43.77	3.03
ReColor	0.2	91.49	**37.6**	**9.06**	37.99	**84.17**	**69.1**	34.02	**18.72**	**51.51**	**38.12**	**68.6**	73.16	31.63	**46.14**	3.02
ReColor	0.5	74.12	34.15	8.14	**41.98**	71.03	58.2	23.92	18.45	43.7	32.07	57.27	55.79	24.93	39.14	**3.53**
Gabor	0	**94.08**	0.3	0.01	0.01	4.43	**92.39**	16.96	8.96	2.08	2.31	17.99	41.61	11.87	16.58	0
Gabor	0.1	91.52	10.06	0.38	**12.04**	**49.59**	89.29	**26.41**	**19.27**	**32.58**	23.99	50.29	54.77	16.1	32.06	0.13
Gabor	0.2	88.56	9.45	0.34	0.72	23.46	85.27	20.59	8.44	17.77	19.99	49	54.96	15.62	25.47	0.04
Gabor	0.5	83.09	**31.81**	**4.79**	1.81	26.15	79.32	20.61	10.17	30.49	**34.34**	**64.64**	**64.17**	**19.19**	**32.29**	**0.44**
Snow	0	**95.89**	0.05	0	0.01	2.63	30.13	**92.02**	7.23	0.9	15.17	31.49	47.98	**22.39**	20.83	0
Snow	0.1	91.74	9.94	0.1	0.51	19.93	69.04	82.63	7.67	12.96	34.24	44.21	70.07	15.53	30.57	0
Snow	0.2	89.18	19.18	0.57	0.95	18.26	**73.07**	76.98	9.2	19.02	36.91	**44.69**	**70.51**	14.24	**31.96**	0.04
Snow	0.5	74.52	**24.82**	**1.68**	**3.67**	**33.1**	57.71	65	**10.01**	**22.39**	**40**	40.36	61.17	21.48	31.78	**0.56**
Pixel	0	**94.76**	0.07	0	0.01	8.87	57.65	**36.6**	**88.35**	1.77	14.52	38.18	67.5	16.21	27.48	0
Pixel	0.1	86.83	0.9	0	1.1	**41.32**	67.39	32.7	77.05	6.22	15.43	42.24	51.45	18.18	29.5	0
Pixel	0.2	89.6	4.25	0.01	0.14	20.42	**68.55**	29.56	76.23	3.57	25.36	**46.03**	**68.2**	19.96	**30.19**	0
Pixel	0.5	61.01	**23.09**	**3.41**	**6.93**	34.6	49.26	19.29	44.75	**17.11**	**30.67**	44.62	52.42	**28.13**	29.52	**0.89**
JPEG	0	**90.26**	56.48	21.5	0.52	34.74	68.59	21.12	10.57	73.46	40	74.3	**78.35**	**28.02**	42.3	0.09
JPEG	0.1	89.2	58.97	25.22	2.37	38.03	**72.82**	22.11	12.85	74.03	41.87	**76.66**	77.74	25.58	44.02	0.76
JPEG	0.2	88.78	**61.28**	**27.5**	8.27	43.35	71.66	22.78	14.46	**74.65**	**43.63**	75.89	78	26.09	45.63	2.01
JPEG	0.5	87.31	59.83	27.21	**15.54**	**45.36**	71.31	**23.82**	**17.46**	73.08	42.7	76.09	76.9	23.25	**46.05**	**3.92**
Elastic	0	**94.06**	1.32	0.02	7.5	7.92	25.41	**53.68**	9.16	11.2	**79.47**	**72.94**	50.24	33.1	29.33	0.01
Elastic	0.1	85.66	13.24	0.91	5.6	14.65	46.65	37.92	5.34	18.66	63.35	70.45	58.79	27.89	30.29	0.39
Elastic	0.2	85.22	28.9	2.8	6.43	18.55	53.96	37.91	6.43	35.93	63.82	69.97	65.53	**33.61**	35.32	0.26
Elastic	0.5	76.85	**42.02**	**10.72**	**26.19**	**46.27**	**58.45**	30.94	**11.09**	**48.24**	57.25	62.24	**65.81**	28.98	**40.68**	**2.76**
Wood	0	**93.57**	0.03	0	0.4	1.27	18.47	**39.44**	3.43	0.37	33.68	**93.04**	28.04	14.11	19.36	0
Wood	0.1	91.08	1.29	0.12	11.65	19.03	49.39	30.26	7.09	10.12	44.63	89.51	50.35	22.37	27.98	0.01
Wood	0.2	90.9	3.9	0.12	**18.97**	**27.03**	50.3	31.63	7.97	18.49	**50.87**	89.35	58.41	22.88	31.66	0.05
Wood	0.5	77.59	**29.6**	**6.65**	9.33	23.37	**55.4**	19.74	**8.3**	**34.75**	49.11	75.26	**59.65**	**24.26**	**32.95**	**0.87**
Glitch	0	**93.26**	0.02	0	0	11.49	49.03	**24.44**	12.47	3.14	10.89	31.99	**90.77**	16.61	20.9	0
Glitch	0.1	85.96	1.29	0.13	0.08	16.46	56.48	19.95	8.38	6.56	20.67	44.52	78.95	18.31	22.65	0
Glitch	0.2	83.4	18.14	1.94	0.23	24.5	**60.76**	19.24	11.85	27.46	33.38	57.36	77.04	**23.12**	29.59	0.01
Glitch	0.5	75.81	**36.99**	**6.88**	**1.22**	**27.29**	59.99	17.29	**14.18**	**34.71**	**36.39**	**60.77**	68.21	22.75	**32.22**	**0.21**
Kaleid-oscope	0	**96.03**	0	0	0	0.8	39.49	**40.94**	5.75	0.02	2.4	43.08	33.71	**91.97**	21.51	0
Kaleid-oscope	0.1	83.96	1.15	0	0.07	19.63	53.87	29.05	7.32	1.72	15.51	**47.8**	**52.33**	72.8	25.1	0
Kaleid-oscope	0.2	82.48	0.8	0	0.07	3.81	**59.93**	25.75	6.5	0.89	10.87	40.35	50.44	66.42	22.15	0
Kaleid-oscope	0.5	46.61	**28.96**	**11.27**	**23.59**	**23.05**	41.04	15.98	**15.01**	**33.14**	**30.43**	42.2	41.4	44.04	**29.18**	**5.31**

Table 17: **Intial Training Ablations- Gaussian regularization on CIFAR-10.** Accuracy of initially trained models on CIFAR-10 trained using different attacks as indicated in "Train Attack" column measured across different attacks. Gaussian regularization (with $\sigma = 0.2$) is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**90.04**	83.95	7.57	5.27	33.91	65.17	**89.3**	28.99	67.52	62.85	49.22	45.78	12.76	46.03	0.51
ℓ_2	0.1	88.97	83.13	12.69	8.54	35.64	66.65	88.2	31.39	68.69	61.73	49.22	46.34	17.3	47.46	1.2
ℓ_2	0.2	89.68	84.36	16.94	9.96	39.95	66.37	88.89	32.48	72.87	**63.67**	52.25	48.64	17.1	49.46	1.83
ℓ_2	0.5	89.76	**84.41**	**25.32**	**16.08**	**44.38**	**69.25**	87.69	**35.46**	**74.42**	62.47	**57.66**	**50.37**	**20.84**	**52.36**	**41.27**
ℓ_∞	0	**84.51**	**81.71**	58.39	43.49	67.82	72.61	**83.31**	41.83	**65.35**	63.9	**67.18**	63.64	30.75	61.67	13.2
ℓ_∞	0.1	82.29	80.25	58.5	47.85	66.8	69.55	81.5	41.4	65.53	61.45	63.13	64.23	33.89	61.17	16.15
ℓ_∞	0.2	83.56	81.04	**59.75**	46.04	67.69	**73.12**	82.75	**42.27**	**67.26**	**63.97**	67.01	62.93	34.04	**62.32**	16.92
ℓ_∞	0.5	81.53	77.91	59.62	**48.48**	**69.3**	67.87	80.23	41.86	54.85	61.71	64.18	**67.77**	**38.98**	61.06	**19.31**
StAdv	0	**83.31**	**77.58**	1.45	**69.81**	**13.43**	36.66	**81.5**	**20.56**	49.89	70.32	60.76	36.15	**24.84**	**45.25**	1.04
StAdv	0.1	81.89	76.31	**2.24**	69.17	12.23	32.23	79.92	18.39	**52.25**	**70.42**	60.46	39.29	20.59	44.46	1.25
StAdv	0.2	82.09	76.66	1.71	69.04	11.11	45.25	80.13	17.1	44.74	69.61	**60.89**	40.03	21.32	44.8	1.25
StAdv	0.5	80.66	73.86	1.68	67.67	10.65	**60.03**	77.58	15.39	44.97	67.8	59.41	**41.86**	21.78	45.22	**1.35**
ReColor	0	91.34	81.53	0.03	0.41	79.08	42.55	90.6	22.39	25.2	64.31	54.8	18.65	8.94	40.71	0
ReColor	0.1	91.26	82.5	**0.05**	0.48	80.74	47.03	91.08	25.3	27.16	63.54	55.39	20.94	11.75	42.16	0
ReColor	0.2	**91.87**	**83.67**	0.03	**0.89**	**81.76**	**49.63**	**91.44**	23.62	**34.62**	**66.37**	57.61	23.64	10.93	43.68	0
ReColor	0.5	90.32	81.43	0	0.84	80.69	46.32	89.55	**30.09**	32.18	65.63	**61.17**	**26.42**	**13.35**	**43.97**	0
Gabor	0	**89.12**	85.27	4.41	2.11	37.83	87.31	**87.62**	20.28	57.66	52.33	38.73	38.88	9.22	43.47	0.1
Gabor	0.1	87.44	83.77	9.45	5.96	40.05	86.01	85.5	21.94	58.47	52.41	41.61	43.57	16.13	45.41	1.25
Gabor	0.2	88.56	**85.48**	13.4	8.84	44.28	**87.36**	87.54	27.77	62.98	54.98	47.92	48.05	**16.87**	48.79	1.4
Gabor	0.5	87.06	84.33	**21.04**	**15.34**	**50.45**	85.38	84.18	**32.05**	**64.84**	**55.62**	**53.61**	**51.44**	15.46	**51.14**	**4.05**
Snow	0	87.69	71.59	**0.08**	1.53	11.97	30.68	62.9	7.31	8.59	66.24	70.78	11.82	9.91	29.45	**0.05**
Snow	0.1	**88.18**	70.47	**0.08**	1.32	11.9	29.86	59.97	6.96	7.26	**66.57**	71.24	11.67	8.61	28.83	0.03
Snow	0.2	87.69	**71.85**	0.03	1.81	12.71	**44.33**	62.27	8.31	9.22	66.39	71.08	**15.54**	**13.12**	**31.39**	0.03
Snow	0.5	88	71.39	0.05	**2.06**	**15.77**	38.88	**62.32**	**10.22**	**10.24**	66.34	**72.25**	15.13	10.47	31.26	0
Pixel	0	88.64	67.24	0	0.59	34.96	30.37	87.18	78.6	0.25	61.63	52.08	49.38	23.97	40.52	0
Pixel	0.1	89.73	69.17	0	0.79	39.18	**37.81**	88.61	81.48	0.51	62.17	49.02	50.11	26.32	42.1	0
Pixel	0.2	**90.47**	**70.55**	0	0.61	45.02	32.69	**89.07**	**83.06**	**1.07**	**65.99**	51.59	53.32	28.23	**43.43**	0
Pixel	0.5	88.2	64.38	0	**1.12**	**45.68**	30.11	87.59	82.37	0.74	64.13	**53.35**	**59.46**	**30.27**	43.27	0
JPEG	0	**88.43**	85.63	15.29	5.43	41.78	77.35	**86.85**	23.21	80.87	53.81	43.39	44.79	15.39	47.82	0.74
JPEG	0.1	88.23	85.68	22.98	9.17	45.22	84.03	86.17	24.08	81.76	**56.87**	46.52	46.47	15.08	50.34	2.04
JPEG	0.2	88.2	**85.71**	25.68	11.82	44.31	82.85	86.17	24.71	81.86	56.25	47.67	47.29	17.66	51	2.55
JPEG	0.5	87.08	84.89	**31.8**	**15.75**	**51.36**	**84.46**	84.92	**28.82**	**82.06**	54.93	**48.18**	**54.85**	**18.6**	**53.38**	**3.8**
Elastic	0	89.66	77.48	0	0.82	12.25	21.81	88.05	16.84	5.71	78.6	62.01	16.79	11.69	32.67	0
Elastic	0.1	**90.96**	80.36	0	2.57	12.54	28.05	**89.71**	16.18	14.04	82.62	65.58	16.99	**14.47**	35.26	0
Elastic	0.2	88.41	79.18	**0.05**	5.15	13.04	26.42	86.7	17.89	15.24	81.27	66.11	**21.02**	13.55	35.47	**0.03**
Elastic	0.5	89.53	**80.79**	0.03	**8.79**	**16.71**	**29.66**	87.62	**19.52**	**22.29**	**83.97**	**66.88**	20.31	11.21	**37.31**	0
Wood	0	85.91	50.09	0	0.82	11.11	35.87	83.31	11.13	1.2	60.94	78.83	14.96	**11.13**	29.95	0
Wood	0.1	88.28	**75.77**	0	2.42	**14.14**	38.9	86.09	**12.97**	**9.94**	67.18	84.31	**22.09**	9.66	**35.29**	0
Wood	0.2	**89.45**	72.61	**0.03**	1.91	11.44	**43.69**	**87.36**	8.05	9.86	**67.49**	**86.88**	16.79	10.17	34.69	**0.03**
Wood	0.5	85.99	67.18	**0.03**	**4.1**	12.87	38.8	83.01	10.34	9.91	66.37	85.07	20.46	9.32	33.96	**0.03**
Glitch	0	**88.51**	36.41	0	0	6.7	18.47	**86.96**	17.1	0	60	50.93	84.97	6.37	30.66	0
Glitch	0.1	88.33	32.38	0	0.08	7.26	21.86	86.27	15.36	0.03	61.2	50.6	**86.68**	6.73	30.7	0
Glitch	0.2	87.85	48.89	**0.03**	0.43	22.78	23.75	85.35	20.08	2.52	63.13	50.34	86.19	10.42	34.49	0
Glitch	0.5	86.98	**65.1**	0	**2.57**	**23.11**	**32.18**	84.05	**30.7**	**7.11**	**64.36**	**58.75**	85.35	**16.64**	**39.16**	0
Kaleid-oscope	0	88.1	73.5	0	0.31	7.03	28.66	85.91	18.83	2.22	62.98	29.78	21.07	84.89	34.6	0
Kaleid-oscope	0.1	88.38	78.98	0.03	2.68	13.96	**33.53**	**85.96**	25.12	11.87	67.13	49.12	27.77	85.66	40.15	0
Kaleid-oscope	0.2	**88.51**	78.78	0.03	6.5	20.48	31.95	85.66	29.4	10.22	68.51	34.85	36.74	**86.09**	40.77	0.03
Kaleid-oscope	0.5	87.01	**79.62**	**0.56**	**16.38**	**24.13**	31.67	84.48	**32.74**	**23.06**	**68.61**	**58.09**	**37.94**	84.2	**45.12**	**0.43**

Table 18: **Intial Training Ablations- worst-case ℓ_2 regularization on ImageNette.** Accuracy of initially trained models on ImageNette trained using different attacks as indicated in "Train Attack" column measured across different attacks. ℓ_2 regularization computed using single step optimization is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**90.04**	83.95	7.57	5.27	33.91	**65.17**	**89.3**	28.99	67.52	62.85	49.22	**45.78**	12.76	46.03	0.51
ℓ_2	0.05	88.51	82.75	13.55	10.14	33.5	63.03	87.11	29.66	69.17	**63.9**	53.27	41.78	17.12	47.08	2.22
ℓ_2	0.1	89.5	83.64	16.89	9.2	41.71	56.97	86.93	**37.91**	71.52	62.62	**54.34**	**51.85**	**20.71**	49.52	1.45
ℓ_2	0.2	88.87	**84.17**	**25.27**	**14.04**	**44.71**	74.27	86.96	36.74	**74.39**	63.36	53.07	49.45	15.34	**51.82**	**2.42**
ℓ_∞	0	**84.51**	**81.71**	58.39	43.49	67.82	**72.61**	**83.31**	41.83	**65.35**	**63.9**	**67.18**	63.64	30.75	**61.67**	13.2
ℓ_∞	0.05	83.13	80.15	58.73	44.92	67.44	65.53	83.24	41.53	61.58	62.42	62.98	65.07	29.63	60.26	14.55
ℓ_∞	0.1	83.36	80.23	58.93	43.57	67.85	69.71	82.85	40.13	56.79	63.13	65.58	65.66	31.97	60.53	16.08
ℓ_∞	0.2	81.22	77.83	**59.21**	**51.57**	**67.97**	65.66	80.69	**42.52**	58.19	61.76	61.15	**67.44**	**40.03**	61.17	**19.13**
StAdv	0	**83.31**	**77.58**	1.45	69.81	13.43	36.66	**81.5**	20.56	49.89	**70.32**	60.76	36.15	24.84	45.25	1.04
StAdv	0.05	82.14	77.17	2.06	73.02	24.94	45.5	78.62	**22.09**	48.97	69.99	62.47	**38.73**	**28.79**	**47.7**	1.63
StAdv	0.1	79.08	72.79	1.86	70.19	**35.87**	48.92	74.52	20.25	43.9	63.29	57.96	35.21	24.05	45.73	1.55
StAdv	0.2	80.89	74.09	**3.31**	**75.49**	30.42	**51.21**	77.61	16.56	**52.56**	67.06	**62.75**	36.28	22.39	47.48	**2.98**
ReColor	0	91.34	81.53	0.03	0.41	79.08	42.55	90.6	22.39	25.2	64.31	54.8	18.65	8.94	40.71	0
ReColor	0.05	91.08	80.64	0.03	0.56	79.24	46.8	90.45	29.3	27.24	64.03	55.31	26.55	12.03	42.68	0
ReColor	0.1	**92.18**	**84.33**	0.18	1.61	82.93	53.66	91.11	29.66	40.79	66.04	57.25	31.16	**15.24**	46.16	0
ReColor	0.2	92.1	83.92	**0.25**	**2.45**	**83.9**	**54.52**	**91.54**	**33.73**	**43.26**	**67.57**	**57.83**	**37.94**	12.51	**47.45**	**0.1**
Gabor	0	**89.12**	**85.27**	4.41	2.11	37.83	87.31	87.62	20.28	**57.66**	52.33	38.73	38.88	9.22	43.47	0.1
Gabor	0.05	88.76	84.48	5.61	3.54	35.49	**87.54**	**88**	18.73	56.61	**56.08**	43.21	37.45	12.69	44.12	0.61
Gabor	0.1	87.92	84.1	8.36	4.79	41.81	85.66	84.92	26.29	54.29	52.97	45.17	45.2	11.08	45.39	0.36
Gabor	0.2	87.97	84.33	**12.46**	**8.99**	**44.36**	86.55	85.27	**29.12**	57.4	55.67	**51.41**	**49.91**	**13.25**	**48.23**	**1.55**
Snow	0	**87.69**	**71.59**	0.08	1.53	**11.97**	30.68	62.9	7.31	8.59	**66.24**	70.78	11.82	9.91	**29.45**	0.05
Snow	0.05	86.85	70.34	0.05	1.48	10.01	**33.2**	**64.71**	7.52	**9.15**	64.46	69.55	11.41	8.94	29.24	0.03
Snow	0.1	86.04	70.96	**0.18**	**1.86**	10.57	23.77	61.43	**8.48**	7.49	65.32	70.34	**15.21**	7.77	28.62	**0.13**
Snow	0.2	87.62	68.89	0.15	1.61	**11.97**	29.91	64.38	7.06	8.03	66.09	**71.8**	12	**9.96**	29.32	0.1
Pixel	0	88.64	67.24	0	0.59	34.96	30.37	87.18	78.6	0.25	61.63	**52.08**	49.38	23.97	40.52	0
Pixel	0.05	89.07	68.99	0	**1.12**	40.66	29.38	87.92	80.79	0.64	62.32	49.32	52.05	25.86	41.59	0
Pixel	0.1	89.2	**70.73**	0	1.04	43.49	**33.71**	88.1	82.55	0.79	**64.48**	49.78	49.94	29.3	42.83	0
Pixel	0.2	**90.7**	67.41	0	1.02	**46.45**	33.25	**89.25**	**83.95**	**1.73**	63.77	50.8	**52.89**	**31.24**	**43.48**	0
JPEG	0	**88.43**	**85.63**	15.29	5.43	41.78	77.35	**86.85**	23.21	**80.87**	53.81	43.39	44.79	15.39	47.82	0.74
JPEG	0.05	87.31	84.54	23.72	8.51	**46.7**	80.56	86.11	24.82	**80.87**	54.39	45.02	45.3	17.32	49.82	2.01
JPEG	0.1	86.75	84.05	24.33	11.39	43.13	79.8	84.89	24.15	80.23	52.79	47.34	47.82	16.36	49.69	2.7
JPEG	0.2	86.65	83.64	**27.57**	**15.44**	45.61	**81.17**	84.74	**26.78**	80.64	**57.68**	**49.99**	**50.93**	**20.08**	**52.02**	**4.13**
Elastic	0	**89.66**	**77.48**	0	0.82	12.25	21.81	88.05	**16.84**	**5.71**	78.6	62.01	**16.79**	11.69	**32.67**	0
Elastic	0.05	88.99	75.67	0	0.74	11.11	19.06	88.23	15.34	5.17	79.21	59.97	15.69	7.85	31.5	0
Elastic	0.1	89.43	**77.48**	0	**1.12**	10.14	25.07	**88.43**	12.43	5.53	**79.44**	**62.09**	13.38	8.82	31.99	0
Elastic	0.2	88.36	74.14	0	0.66	**12.33**	**26.93**	87.31	15.13	3.13	77.25	58.98	14.75	**13.99**	32.05	0
Wood	0	85.91	50.09	0	0.82	11.11	**35.87**	83.31	**11.13**	1.2	60.94	78.83	14.96	**11.13**	29.95	0
Wood	0.05	**88.61**	72.03	**0.03**	0.36	10.27	29.43	86.11	8.43	4.1	**67.69**	**85.27**	17.99	9.76	32.62	0
Wood	0.1	87.54	70.93	**0.03**	0.48	**12.25**	23.59	**86.62**	10.6	5.2	65.1	83.24	**20.87**	8	32.24	**0.03**
Wood	0.2	87.77	**73.89**	0	**2.22**	10.14	34.78	86.04	9.15	**9.27**	66.14	84.08	20.28	11.03	**33.92**	0
Glitch	0	**88.51**	36.41	0	0	6.7	18.47	**86.96**	17.1	0	60	50.93	84.97	6.37	30.66	0
Glitch	0.05	87.97	6.62	0	0	4.2	**29.81**	86.5	12.1	0	53.71	49.12	81.91	2.96	27.24	0
Glitch	0.1	87.52	31.75	0	0.05	16.1	21.25	86.06	19.21	0.05	**62.62**	**55.01**	**86.24**	6.04	32.03	0
Glitch	0.2	86.47	**53.61**	0	**0.64**	**23.21**	24.51	85.22	**25.55**	**1.35**	61.15	52.08	84.79	**9.86**	**35.16**	0
Kaleid-oscope	0	**88.1**	73.5	0	0.31	7.03	28.66	85.91	18.83	2.22	**62.98**	29.78	21.07	84.89	34.6	0
Kaleid-oscope	0.05	86.9	71.03	**0.03**	0.64	6.24	**29.15**	85.32	15.69	3.11	59.85	**44.71**	23.49	84.23	**35.29**	0
Kaleid-oscope	0.1	87.64	74.14	0	0.56	9.15	25.71	85.4	17.58	**3.87**	60.87	27.52	**25.81**	85.15	34.65	0
Kaleid-oscope	0.2	87.46	**74.8**	0	**1.25**	**9.78**	24.36	**86.06**	**20.87**	3.49	62.39	28.15	23.64	**85.17**	35	0

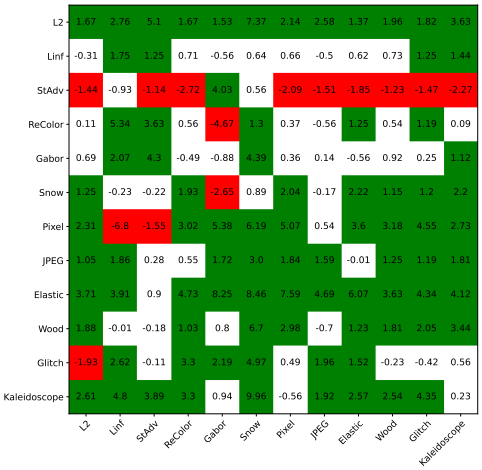
Table 19: **Intial Training Ablations- variation regularization on ImageNette.** Accuracy of initially trained models on ImageNette trained using different attacks as indicated in "Train Attack" column measured across different attacks. Variation regularization computed using single step optimization is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid -oscope	Avg	Union
ℓ_2	0	**90.04**	**83.95**	7.57	5.27	33.91	65.17	**89.3**	28.99	67.52	62.85	49.22	**45.78**	12.76	46.03	0.51
ℓ_2	0.5	89.68	83.49	8.64	6.17	34.17	64.59	88.92	29.1	68.41	**64.59**	**51.49**	42.14	13.58	46.27	0.36
ℓ_2	1	89.45	82.9	10.37	**7.49**	33.1	**67.95**	87.59	28.05	**69.38**	61.27	50.83	42.5	**14.9**	**46.36**	**0.76**
ℓ_2	2	89.76	83.36	**10.6**	6.73	**35.01**	63.03	88.46	**29.17**	68.79	62.83	51.11	42.22	12.87	46.18	0.61
ℓ_∞	0	84.51	81.71	58.39	43.49	67.82	**72.61**	83.31	41.83	65.35	63.9	**67.18**	63.64	30.75	61.67	13.2
ℓ_∞	0.5	84.51	81.63	58.93	44.66	67.62	72.05	83.69	43.31	62.62	62.22	64.54	66.42	32.71	61.7	14.42
ℓ_∞	1	83.97	81.22	**59.01**	44.87	66.6	71.9	81.73	43.52	66.47	63.26	66.39	**66.73**	**33.99**	62.14	**15.97**
ℓ_∞	2	**85.2**	**83.36**	58.42	**47.92**	**69.27**	69.99	**84.18**	**43.97**	**73.53**	**64.33**	66.7	62.75	26.75	**62.6**	13.81
StAdv	0	83.31	77.58	1.45	**69.81**	13.43	36.66	81.5	20.56	49.89	70.32	60.76	36.15	24.84	45.25	1.04
StAdv	0.5	84.1	79.31	2.09	**69.81**	16.28	39.52	81.55	**23.95**	55.97	71.49	62.6	**42.42**	**27.95**	**47.75**	1.5
StAdv	1	**84.89**	**79.67**	2.52	68.64	14.27	**41.73**	**82.27**	21.17	56.59	71.49	**63.64**	37.81	24.03	46.99	**1.81**
StAdv	2	83.62	79.34	**2.96**	69.66	**16.59**	40.08	81.66	22.06	**59.87**	**71.54**	62.01	39.18	21.66	47.22	1.68
ReColor	0	91.34	81.53	0.03	0.41	79.08	42.55	90.6	22.39	25.2	64.31	54.8	18.65	8.94	40.71	0
ReColor	0.5	91.82	82.85	0.08	0.38	79.75	50.83	91.08	22.83	28.28	62.57	53.17	21.48	9.91	41.93	0
ReColor	1	91.77	85.2	0.15	0.41	**81.2**	**54.22**	91.39	**27.08**	38.32	**64.82**	**55.08**	**23.06**	**10.65**	**44.3**	0
ReColor	2	**92.31**	**85.53**	**0.28**	**0.54**	80.99	52.69	**91.67**	24.46	**44.38**	**64.82**	52.61	21.94	9.81	44.14	**0.03**
Gabor	0	**89.12**	**85.27**	4.41	2.11	**37.83**	**87.31**	**87.62**	**20.28**	57.66	52.33	38.73	38.88	9.22	**43.47**	0.1
Gabor	0.5	88.33	84.1	4.41	2.9	33.4	86.75	86.06	18.62	54.29	**53.68**	35.49	36.33	10.78	42.24	0.13
Gabor	1	88.23	84.41	6.57	4.03	31.64	85.43	87.24	18.09	57.12	53.4	**40.2**	38.09	11.11	43.11	0.71
Gabor	2	86.19	82.39	**9.12**	**5.91**	33.1	84.03	84.38	17.68	**63.16**	44.89	33.1	**43.67**	**14.32**	42.98	**0.84**
Snow	0	87.69	71.59	0.08	1.53	11.97	30.68	**62.9**	7.31	8.59	66.24	**70.78**	11.82	**9.91**	29.45	0.05
Snow	0.5	**88.46**	76.46	0.03	1.12	13.22	36.94	58.78	8.84	10.96	66.42	70.62	18.27	6.42	30.67	0.03
Snow	1	**87.82**	77.04	0.1	1.48	14.14	43.21	59.06	8.64	13.48	65.22	68.89	17.76	7.67	31.39	0.08
Snow	2	88.36	**79.29**	**0.15**	**2.34**	**16.2**	**48.99**	59.69	**10.14**	**21.48**	**68.13**	70.24	**19.03**	9.58	**33.77**	**0.1**
Pixel	0	88.64	67.24	0	0.59	34.96	30.37	87.18	78.6	0.25	61.63	**52.08**	49.38	23.97	40.52	0
Pixel	0.5	**89.81**	75.54	0	0.43	38.8	31.06	87.67	**80.48**	1.1	**63.67**	49.58	**50.01**	23.26	41.8	0
Pixel	1	89.43	**75.85**	0	0.84	**39.95**	**34.52**	**88**	80.25	1.27	63.18	49.27	46.96	21.78	**41.82**	0
Pixel	2	87.62	75.62	0	**1.27**	39.03	30.22	86.04	77.86	**2.88**	61.15	**52.08**	49.3	**24.1**	41.63	0
JPEG	0	88.43	85.63	15.29	5.43	41.78	77.35	86.85	23.21	80.87	53.81	**43.39**	44.79	15.39	47.82	0.74
JPEG	0.5	**89.02**	**86.17**	**17.89**	6.52	**43.29**	**79.01**	**87.52**	**24.56**	**81.15**	55.62	42.78	**45.55**	**15.92**	**48.83**	1.3
JPEG	1	87.9	84.89	16.59	7.34	40.13	78.06	86.65	21.94	80.38	55.57	42.93	44.48	15.77	47.89	**1.66**
JPEG	2	88.08	85.1	17.68	**7.46**	41.58	73.02	86.29	23.49	79.85	**56.51**	42.8	45.25	13.3	47.69	0.97
Elastic	0	89.66	77.48	0	0.82	12.25	21.81	88.05	16.84	5.71	78.6	62.01	16.79	11.69	32.67	0
Elastic	0.5	90.68	81.61	**0.03**	1.66	14.11	**35.57**	89.63	**18.27**	11.87	80.64	60.61	18.45	**13.76**	35.52	**0.03**
Elastic	1	**91.13**	82.93	0	1.78	14.37	31.69	**89.99**	17.07	17.4	81.48	**65.1**	**21.53**	10.96	36.19	0
Elastic	2	90.06	**82.96**	**0.03**	**1.86**	**16.54**	31.75	89.1	17.22	**20.38**	**81.53**	64.61	18.88	10.42	**36.27**	0
Wood	0	85.91	50.09	0	0.82	11.11	35.87	83.31	11.13	1.2	60.94	78.83	14.96	11.13	29.95	0
Wood	0.5	88.03	74.22	0	0.97	11.77	41.02	85.68	9.81	6.57	64.54	81.76	**18.24**	11.01	33.8	0
Wood	1	88.82	78.68	0.03	0.69	14.06	40.46	86.65	9.83	11.49	66.7	**84.94**	17.02	9.76	35.03	0
Wood	2	**89.38**	**81.71**	**0.31**	**2.62**	**15.26**	**43.64**	**87.64**	**11.16**	**24.64**	**68.99**	84.84	17.81	**12.56**	**37.6**	**0.18**
Glitch	0	88.51	36.41	0	0	6.7	18.47	86.96	17.1	0	60	50.93	84.97	6.37	30.66	0
Glitch	0.5	**90.01**	74.73	0	0.54	14.96	24.38	**87.97**	24.69	2.7	64.82	52.92	**86.98**	8	36.89	0
Glitch	1	88.05	76.25	**0.03**	**1.38**	14.5	**28.25**	85.73	25.3	3.41	62.17	**53.1**	84.69	8.79	36.97	0
Glitch	2	89.55	**80.97**	**0.03**	1.25	**15.16**	25.89	87.9	**27.54**	**7.52**	**65.07**	52.05	85.99	**10.47**	**38.32**	0
Kaleid-oscope	0	**88.1**	73.5	0	0.31	7.03	28.66	85.91	18.83	2.22	62.98	29.78	21.07	**84.89**	34.6	0
Kaleid-oscope	0.5	85.2	75.8	0.05	3.01	12.79	47.67	82.45	19.8	15.67	59.18	**41.48**	29.71	81.32	39.08	**0.05**
Kaleid-oscope	1	87.52	**80.48**	0.03	2.52	**20.89**	45.89	**86.04**	23.85	**27.57**	**64.28**	33.63	27.69	83.03	**41.32**	0
Kaleid-oscope	2	80.41	71.54	**0.84**	**5.78**	19.06	**54.55**	76.2	**25.02**	24.41	52.56	27.82	**35.41**	75.49	39.06	0.46

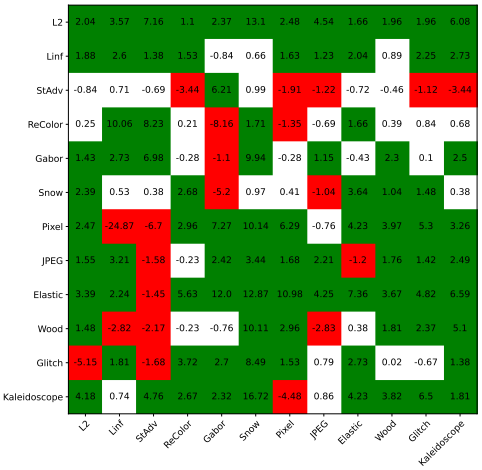
Table 20: **Intial Training Ablations- Uniform regularization on ImageNette.** Accuracy of initially trained models on ImageNette trained using different attacks as indicated in "Train Attack" column measured across different attacks. Uniform regularization (with $\sigma = 2$) is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.

Train Attack	λ	Clean	ℓ_2	ℓ_∞	StAdv	ReColor	Gabor	Snow	Pixel	JPEG	Elastic	Wood	Glitch	Kaleid-oscope	Avg	Union
ℓ_2	0	**90.04**	**83.95**	7.57	5.27	33.91	65.17	**89.3**	28.99	67.52	**62.85**	49.22	45.78	12.76	46.03	0.51
ℓ_2	0.1	89.35	83.13	12.84	6.22	35.59	72.05	87.82	29.45	71.36	62.22	50.14	45.02	16.18	47.67	1.07
ℓ_2	0.2	89.38	83.54	15.82	7.54	38.04	73.43	88.41	30.6	72.2	60.13	50.62	46.14	15.64	48.51	1.35
ℓ_2	0.5	87.03	81.89	**22.09**	**12.31**	**40.36**	**74.09**	84.15	**34.98**	**72.79**	60.13	**51.49**	**52.2**	**18.45**	**50.41**	**2.96**
ℓ_∞	0	**84.51**	81.71	58.39	43.49	67.82	72.61	**83.31**	41.83	65.35	63.9	**67.18**	**63.64**	30.75	61.67	13.2
ℓ_∞	0.1	83.9	82.09	57.86	47.77	68.38	73.4	82.96	44.92	74.55	63.21	64.41	63.18	**36.36**	63.26	17.04
ℓ_∞	0.2	84.25	**82.39**	58.39	**49.38**	**68.59**	**76.33**	82.9	45.45	76.46	**64.56**	66.19	63.36	31.44	**63.79**	16.25
ℓ_∞	0.5	83.87	82.22	**58.42**	47.87	68.23	75.69	82.78	**46.24**	**77.53**	62.14	64.87	62.5	35.31	63.65	**17.66**
StAdv	0	**83.31**	77.58	1.45	**69.81**	13.43	36.66	**81.5**	20.56	49.89	**70.32**	60.76	36.15	24.84	45.25	1.04
StAdv	0.1	82.34	77.76	4.18	66.45	16.54	**57.91**	79.52	22.14	60.43	68.56	**62.29**	40.59	23.46	48.32	2.78
StAdv	0.2	83.11	79.21	6.5	67.64	**18.7**	45.32	80.03	26.34	64.84	70.27	61.94	**50.14**	30.37	50.11	3.34
StAdv	0.5	83.13	**80.03**	**9.35**	68.25	18.34	57.63	80.28	**27.44**	**68.08**	69.78	60.05	49.4	**32.66**	**51.77**	**5.22**
ReColor	0	**91.34**	81.53	0.03	0.41	79.08	42.55	90.6	22.39	25.2	**64.31**	**54.8**	18.65	8.94	40.71	0
ReColor	0.1	90.83	84.97	0.38	0.92	**79.69**	57.17	**90.62**	29.94	48.41	60.94	53.02	33.81	8.54	45.7	0.05
ReColor	0.2	90.8	**85.73**	1.2	1.17	78.98	62.88	90.5	27.87	55.34	60.74	49.07	31.57	14.29	46.61	0.18
ReColor	0.5	89.15	85.2	**5.81**	**2.75**	77.81	**71.26**	88.46	**31.34**	**65.07**	59.77	52.41	**41.27**	**15.39**	**49.71**	**0.33**
Gabor	0	**89.12**	**85.27**	4.41	2.11	37.83	**87.31**	**87.62**	20.28	57.66	52.33	38.73	38.88	9.22	43.47	0.1
Gabor	0.1	87.29	83.77	11.26	5.53	38.22	85.58	84.54	20.64	66.11	48.82	37.02	41.96	15.06	44.87	0.76
Gabor	0.2	86.22	82.88	12.97	6.47	37.27	84.69	83.92	22.45	67.77	48.46	36.31	41.15	19.01	45.28	1.04
Gabor	0.5	87.29	84.13	**19.87**	**9.48**	**41.35**	85.4	85.07	**26.8**	**72.28**	**52.79**	**42.29**	**46.19**	**19.9**	**48.8**	**1.99**
Snow	0	**87.69**	71.59	0.08	1.53	11.97	30.68	**62.9**	7.31	8.59	66.24	70.78	11.82	9.91	29.45	0.05
Snow	0.1	**87.69**	80.25	1.35	5.81	16.41	60.87	57.02	10.09	38.24	65.17	**71.77**	22.62	10.85	36.7	0.38
Snow	0.2	86.8	80.36	1.94	**23.82**	**33.45**	61.55	58.19	11.97	47.77	**67.16**	71.54	25.68	**15.9**	41.61	1.66
Snow	0.5	85.04	**80.46**	**4.31**	15.34	21.55	**66.65**	55.82	**14.04**	**56.33**	65.3	69.48	**32.51**	13.61	**41.28**	**2.6**
Pixel	0	88.64	67.24	0	0.59	34.96	30.37	87.18	78.6	0.25	**61.63**	**52.08**	**49.38**	23.97	40.52	0
Pixel	0.1	88.15	76.99	0.03	0.59	41.5	42.04	85.2	77.25	8.56	56.64	46.88	43.92	28.2	42.32	0
Pixel	0.2	**88.89**	80.99	0.03	1.4	46.7	44.23	**87.64**	**78.73**	20.18	61.4	46.85	49.12	26.37	45.3	0
Pixel	0.5	86.6	**82.14**	**1.43**	**6.62**	**46.73**	**53.45**	84.97	77.35	**44.99**	59.08	48.28	53.73	**28.66**	**48.95**	**0.38**
JPEG	0	88.43	85.63	15.29	5.43	41.78	77.35	86.85	23.21	80.87	53.81	43.39	44.79	15.39	47.82	0.74
JPEG	0.1	88.18	84.79	20.33	9.2	42.34	80.54	85.45	23.36	80.18	**57.38**	46.55	46.78	17.96	49.57	2.22
JPEG	0.2	**88.92**	**86.04**	23.31	8.28	**46.88**	**80.61**	**87.36**	26.73	**81.68**	57.22	47.18	48.94	16.36	50.88	1.86
JPEG	0.5	87.18	84.41	**26.5**	**12.46**	44.94	79.36	85.07	**28.36**	80.54	56.59	**47.77**	**52.71**	**19.01**	**51.48**	**3.06**
Elastic	0	**89.66**	77.48	0	0.82	12.25	21.81	**88.05**	16.84	5.71	78.6	62.01	16.79	11.69	32.67	0
Elastic	0.1	88.59	82.65	0.46	4.31	17.83	45.53	87.44	18.8	35.85	79.85	**64.23**	27.11	11.11	39.6	0.08
Elastic	0.2	89.35	**84.03**	1.66	5.55	**20.64**	53.22	88	22.45	50.75	**80.15**	62.29	31.75	8.1	42.38	0.23
Elastic	0.5	82.17	77.61	**6.9**	**62.17**	63.8	**55.44**	78.32	**29.61**	**55.75**	73.22	63.36	**36.66**	**19.52**	**51.86**	**4.15**
Wood	0	85.91	50.09	0	0.82	11.11	35.87	83.31	11.13	1.2	60.94	78.83	14.96	11.13	29.95	0
Wood	0.1	88.1	80.71	0.87	6.68	14.19	46.55	86.22	11.85	33.3	66.68	82.88	30.8	8.1	39.07	0.33
Wood	0.2	**88.99**	83.03	2.14	7.67	17.17	45.25	**86.98**	13.96	45.22	68.33	**83.67**	30.29	14.19	41.49	1.15
Wood	0.5	88.13	**83.92**	**5.53**	**13.66**	**24.87**	**64.15**	86.68	**18.83**	**60.41**	**69.94**	82.98	**39.9**	**15.77**	**47.22**	**1.99**
Glitch	0	**88.51**	36.41	0	0	6.7	18.47	**86.96**	17.1	0	60	50.93	**84.97**	6.37	30.66	0
Glitch	0.1	87.41	80.08	1.32	4.56	27.77	48.15	85.1	26.93	46.19	61.45	48.56	82.47	12.36	43.75	0.33
Glitch	0.2	86.93	83.54	4.54	8.31	24.08	47.34	85.73	32.94	53.61	**62.29**	**52.36**	83.21	17.4	46.28	1.27
Glitch	0.5	87.16	**84.38**	**13.73**	**14.96**	**28.94**	**62.11**	84.89	**35.9**	**67.92**	60.92	51.46	82.8	**20.48**	**50.71**	**2.8**
Kaleid-oscope	0	**88.1**	**73.5**	0	0.31	7.03	28.66	**85.91**	18.83	2.22	**62.98**	29.78	21.07	**84.89**	34.6	0
Kaleid-oscope	0.1	84.05	72.25	0.1	2.98	14.27	40.74	80.87	14.7	14.29	55.39	**46.42**	21.15	74.96	36.51	0.1
Kaleid-oscope	0.2	69.17	62.19	1.27	10.11	18.7	55.72	66.14	17.81	30.57	41.78	21.68	32.36	66.8	35.43	0.66
Kaleid-oscope	0.5	71.95	66.85	**7.82**	**28.54**	**21.91**	**58.5**	68.64	**24.92**	**47.44**	47.77	38.32	**43.62**	65.04	**43.28**	**6.01**

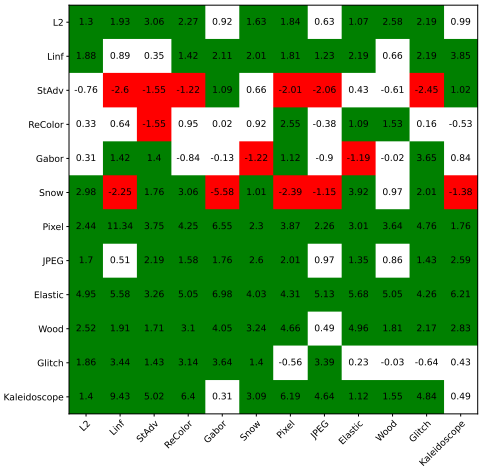
Table 21: **Intial Training Ablations- Gaussian regularization on ImageNette.** Accuracy of initially trained models on ImageNette trained using different attacks as indicated in "Train Attack" column measured across different attacks. Gaussian regularization (with $\sigma = 0.2$) is also considered during initial training, with regularization strength λ. Results where regularization improves over no regularization ($\lambda = 0$) by at least 1% accuracy are highlighted in green, while results where regularization incurs at least a 1% drop in accuracy are highlighted in red. Best performing with respect to regularization strength are bolded.



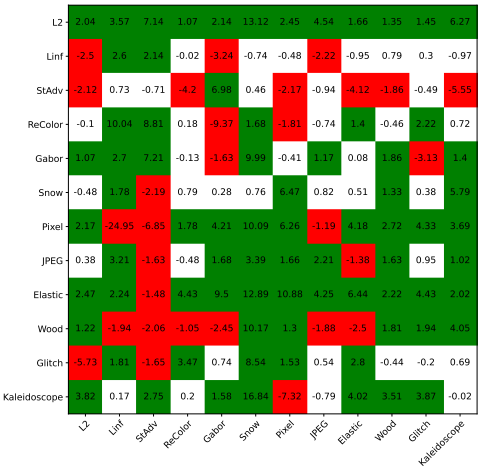
(a) Difference in Avg Acc



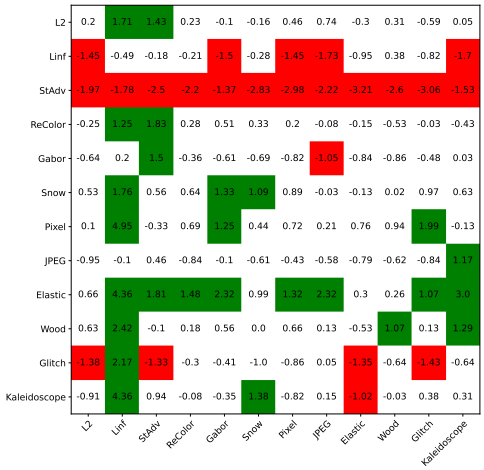
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

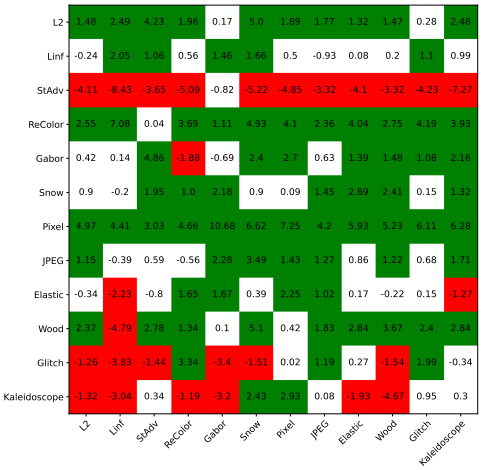


(d) Difference in New Attack Acc

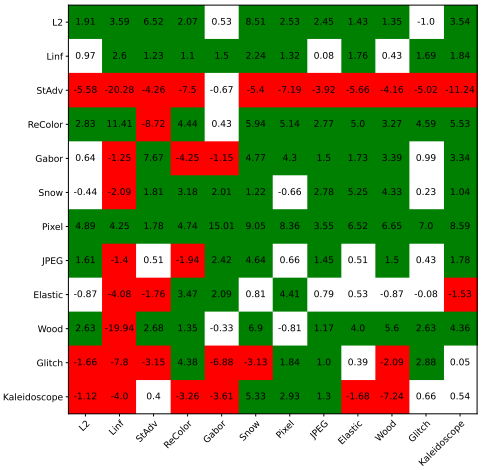


(e) Difference in Clean Acc

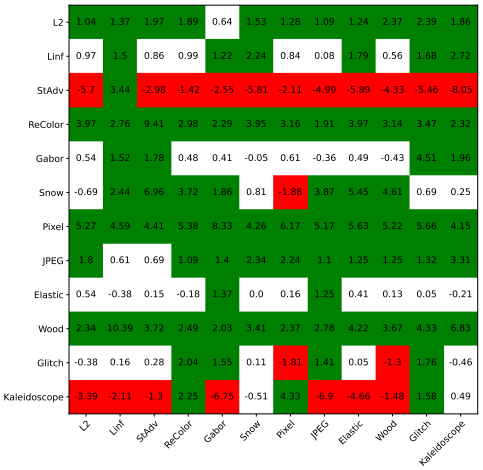
Figure 6: **Change in robust accuracy after fine-tuning with models initally trained with adversarial ℓ_2 regularization different initial attack and new attack pairs.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization in initial training. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



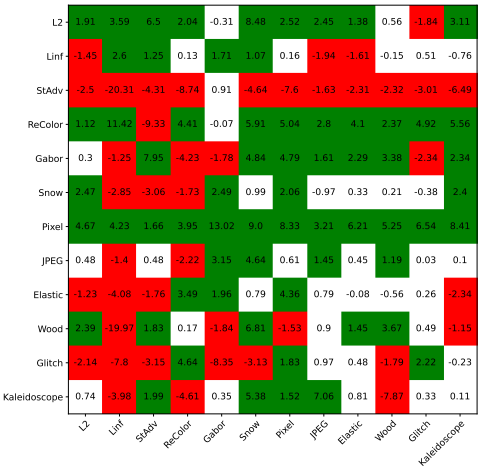
(a) Difference in Avg Acc



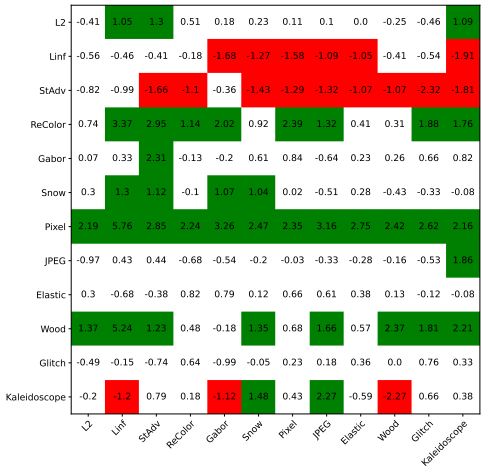
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

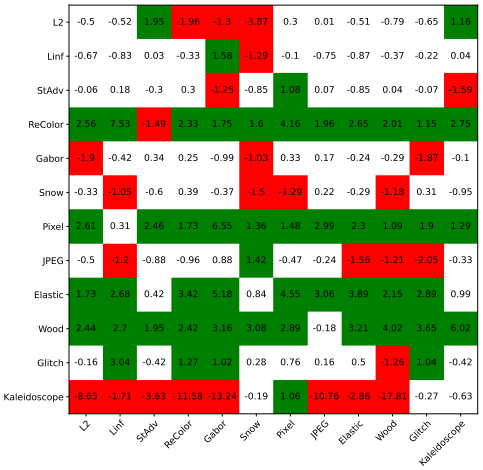


(d) Difference in New Attack Acc

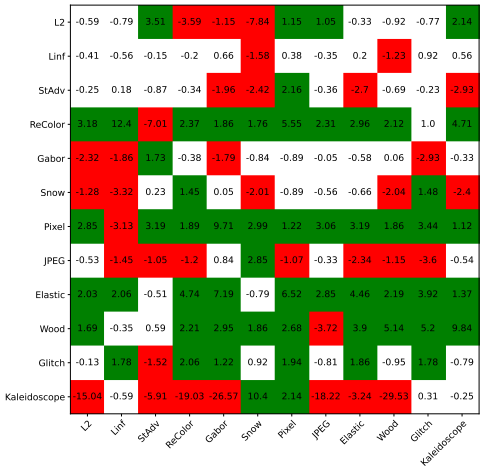


(e) Difference in Clean Acc

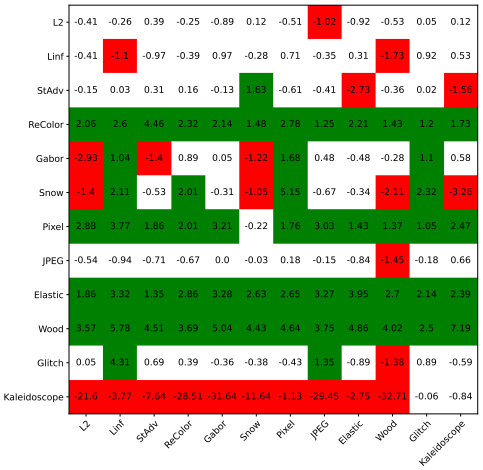
Figure 7: **Change in robust accuracy after fine-tuning with models initally trained with variation regularization different initial attack and new attack pairs.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization in initial training. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



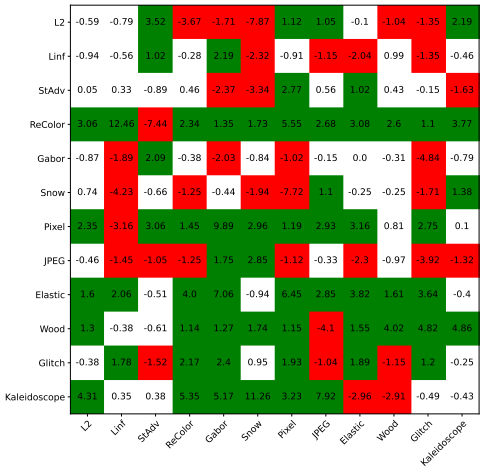
(a) Difference in Avg Acc



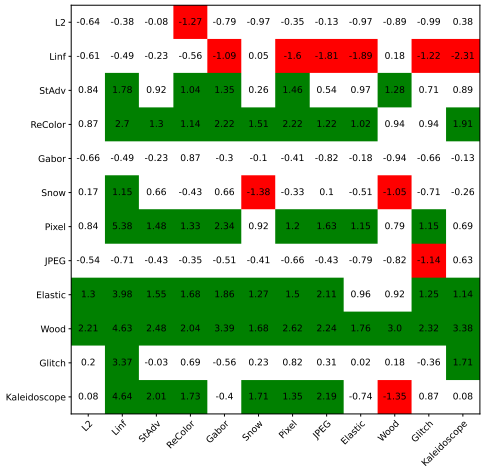
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

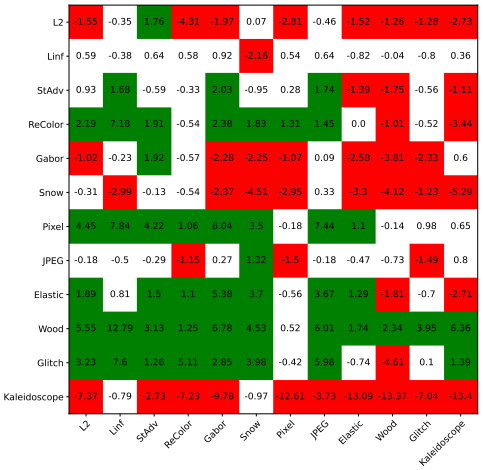


(d) Difference in New Attack Acc

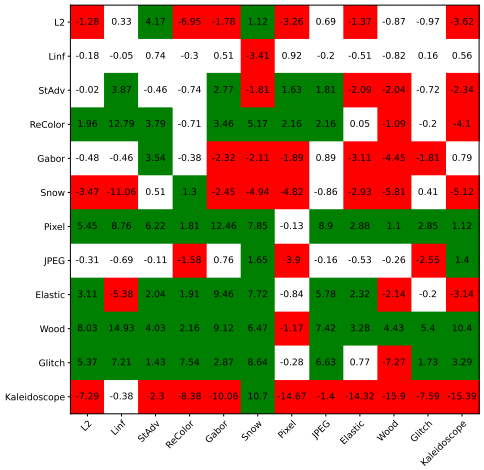


(e) Difference in Clean Acc

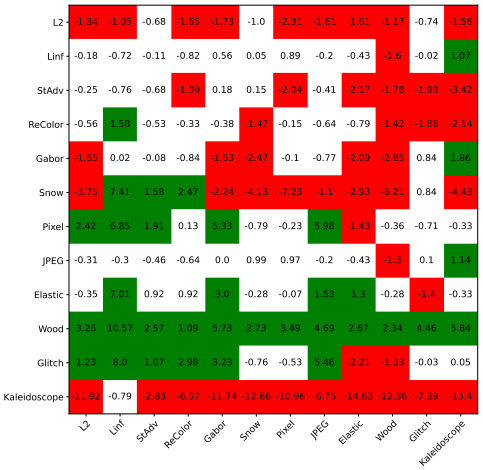
Figure 8: **Change in robust accuracy after fine-tuning with models initally trained with uniform regularization different initial attack and new attack pairs.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization in initial training. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



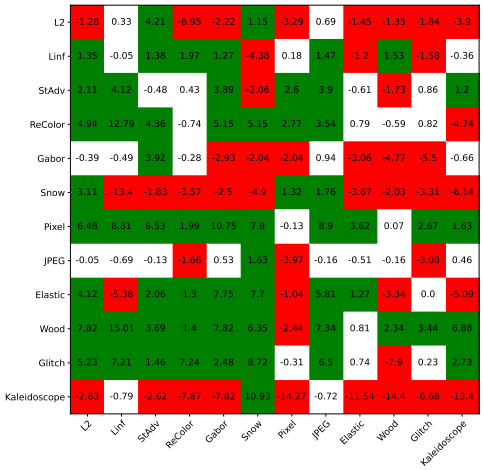
(a) Difference in Avg Acc



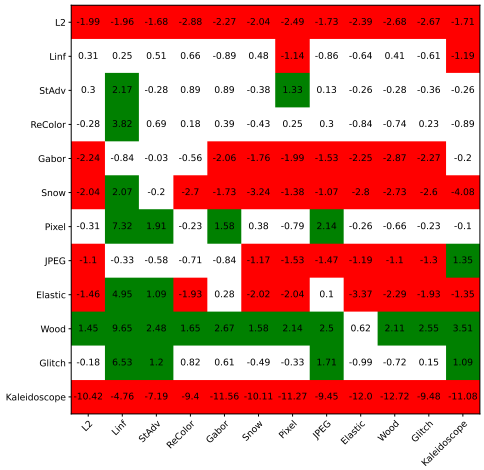
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc



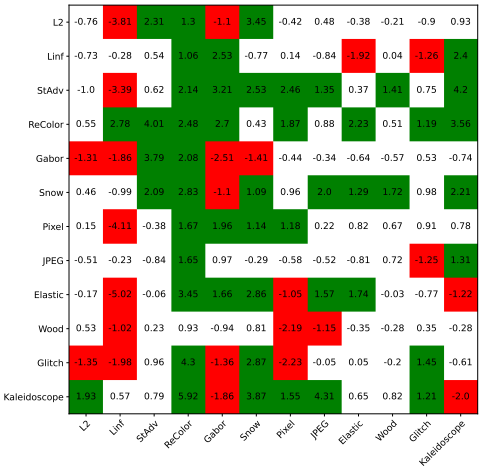
(d) Difference in New Attack Acc



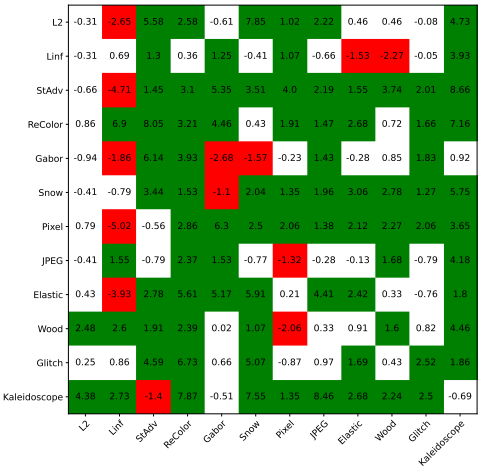
(e) Difference in Clean Acc

Figure 9: **Change in robust accuracy after fine-tuning with models initally trained with Gaussian regularization different initial attack and new attack pairs.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization in initial training. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.

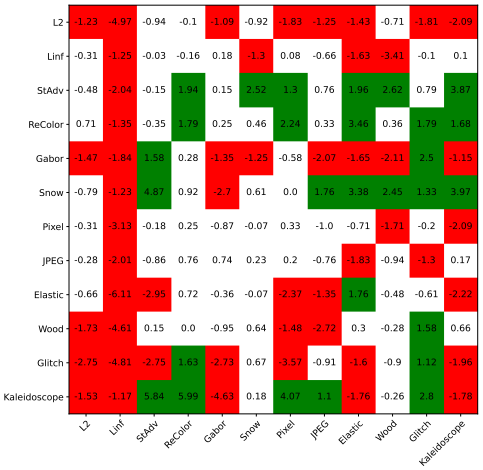
Random noise regularization in fine-tuning hurts overall robustness. Unlike adversarial ℓ_2 regularization which can improve performance when used in both initial training and regularization, we find that uniform and Gaussian regularization generally hurts average, union, initial attack, and new attack accuracies when incorporated in fine-tuning. This suggests that while random noise based regularization may help with initial training (and unforeseen robustness), they do not necessarily help with continual adaptive robustness through fine-tuning.



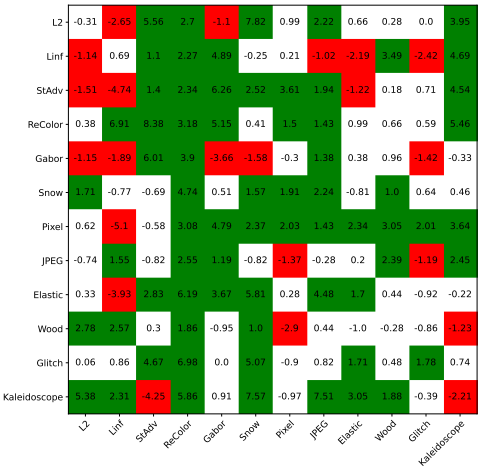
(a) Difference in Avg Acc



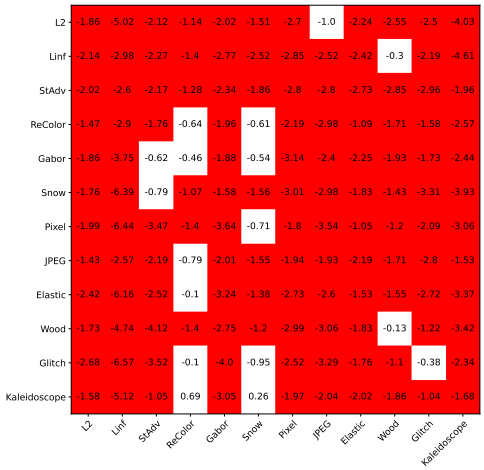
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

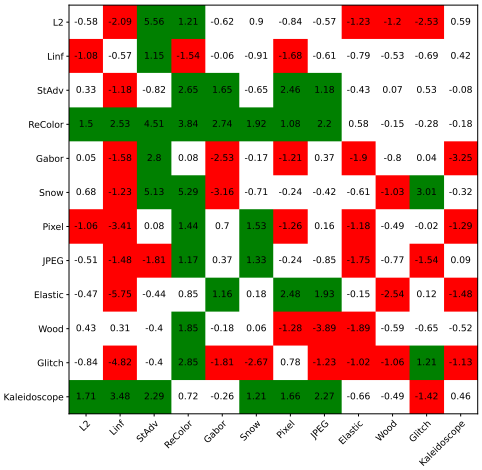


(d) Difference in New Attack Acc

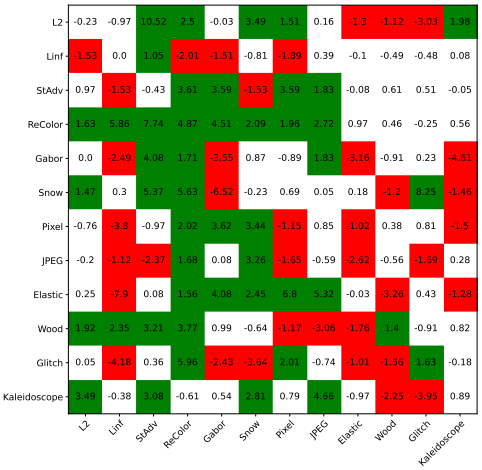


(e) Difference in Clean Acc

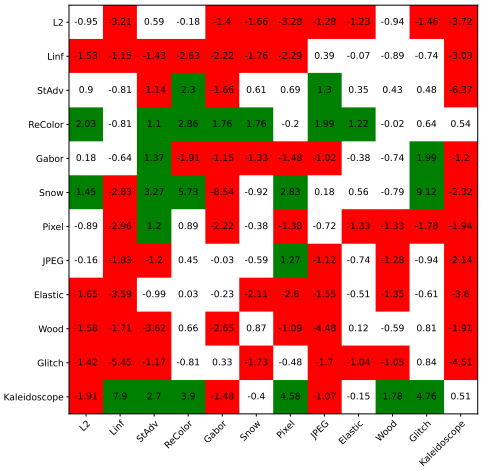
Figure 10: **Change in robust accuracy after fine-tuning with adversarial ℓ_2 regularization.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



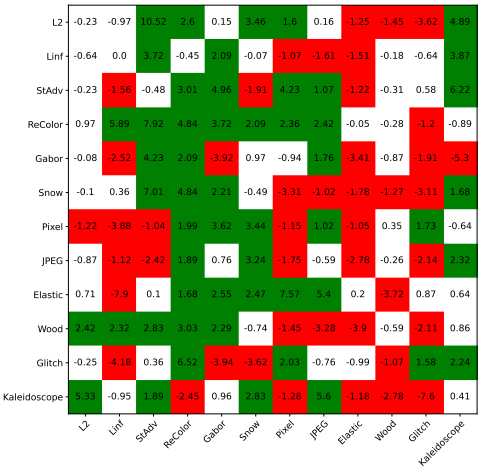
(a) Difference in Avg Acc



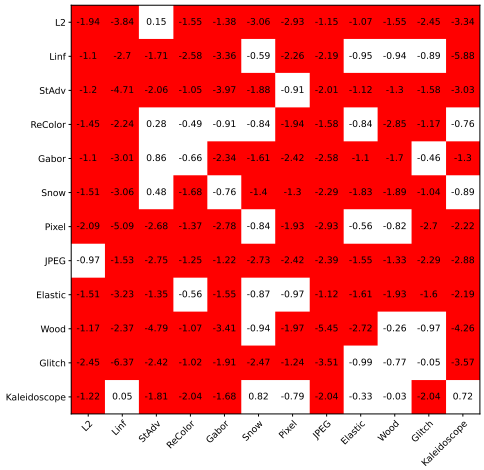
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

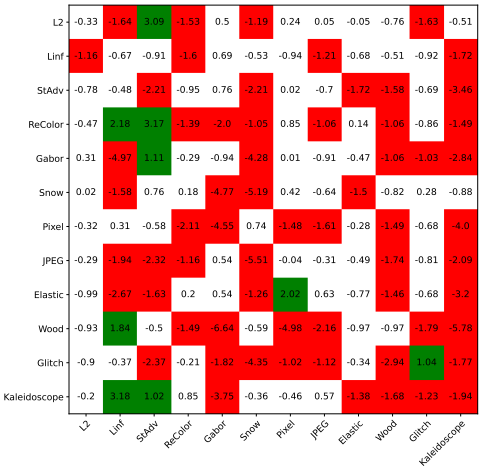


(d) Difference in New Attack Acc

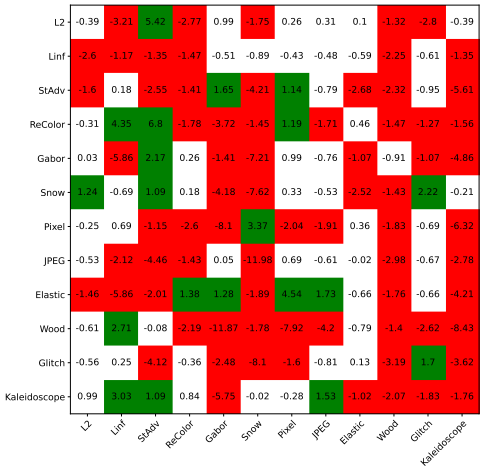


(e) Difference in Clean Acc

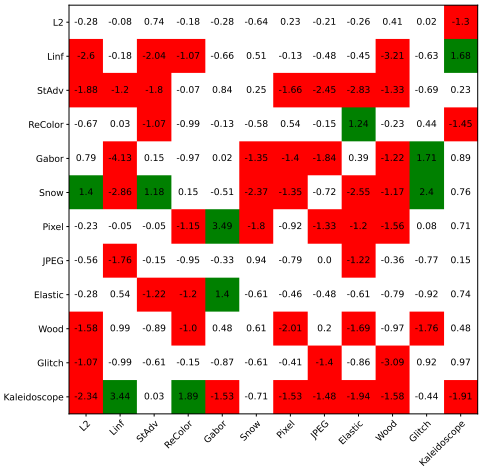
Figure 11: **Change in robust accuracy after fine-tuning with variation regularization.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



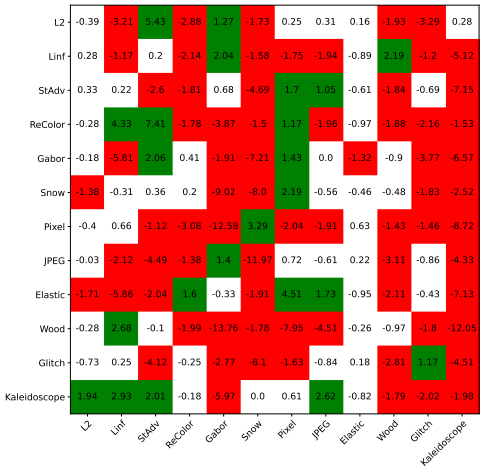
(a) Difference in Avg Acc



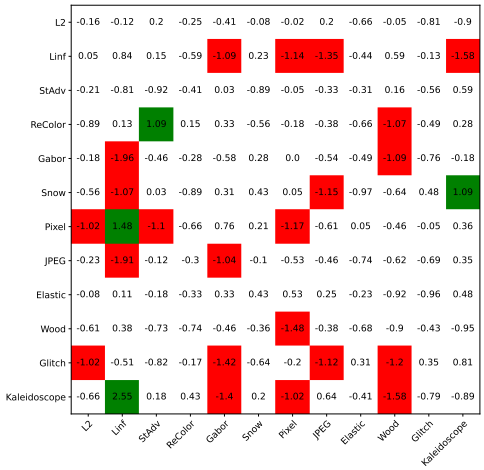
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc

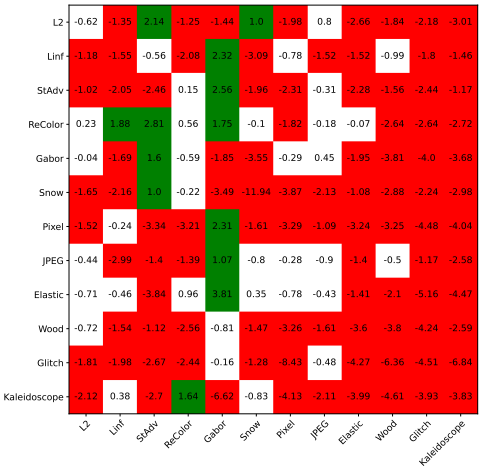


(d) Difference in New Attack Acc

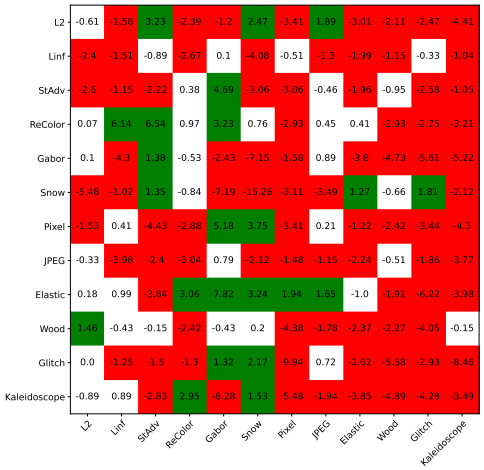


(e) Difference in Clean Acc

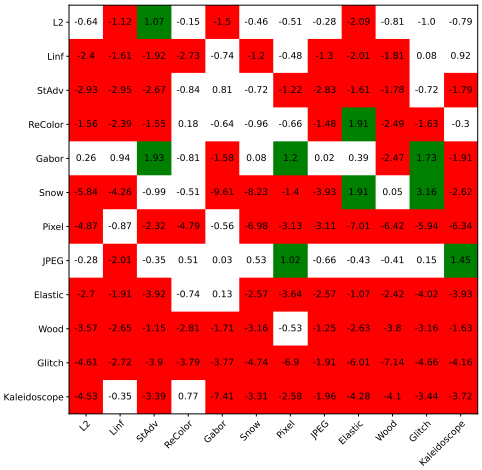
Figure 12: **Change in robust accuracy after fine-tuning with uniform regularization.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.



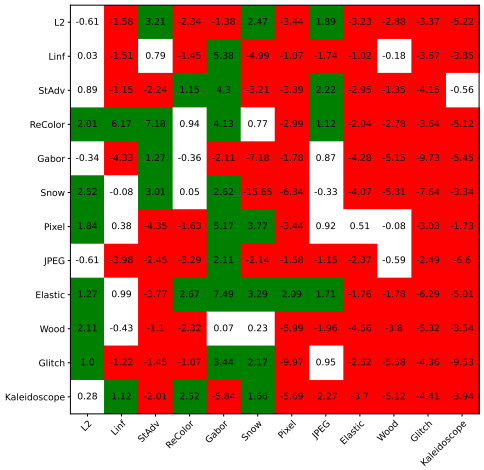
(a) Difference in Avg Acc



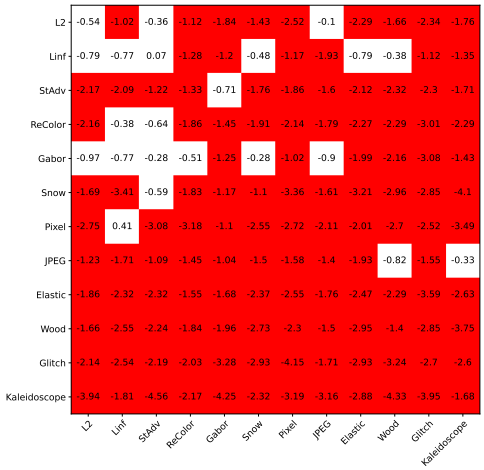
(b) Difference in Union Acc



(c) Difference in Initial Attack Acc



(d) Difference in New Attack Acc



(e) Difference in Clean Acc

Figure 13: **Change in robust accuracy after fine-tuning with Gaussian regularization.** We fine-tune models on Imagenette across 144 pairs of initial attack and new attack. The initial attack corresponds to the row of each grid and new attack corresponds to each column. Values represent differences between the accuracy measured on a model fine-tuned with and without regularization. Gains in accuracy of at least 1% are highlighted in green, while drops in accuracy of at least 1% are highlighted in red.