Sample Efficient Detection and Classification of Adversarial Attacks via Self-Supervised Embeddings

Mazda Moayeri
Department of Computer Science
University of Maryland
mmoayeri@umd.edu

Soheil Feizi
Department of Computer Science
University of Maryland
sfeizi@cs.umd.edu

Abstract

Adversarial robustness of deep models is pivotal in ensuring safe deployment in real world settings, but most modern defenses have narrow scope and expensive costs. In this paper, we propose a self-supervised method to detect adversarial attacks and classify them to their respective threat models, based on a linear model operating on the embeddings from a pre-trained self-supervised encoder. We use a SimCLR encoder in our experiments, since we show the SimCLR embedding distance is a good proxy for human perceptibility, enabling it to encapsulate many threat models at once. We call our method **SimCat** *since it uses SimCLR encoder to catch and categorize various types of adversarial attacks, including ℓ_p and non-ℓ_p evasion attacks, as well as data poisonings. The simple nature of a linear classifier makes our method efficient in both time and sample complexity. For example, on SVHN, using only five pairs of clean and adversarial examples computed with a PGD-ℓ_∞ attack, SimCat's detection accuracy is over 85%. Moreover, on ImageNet, using only 25 examples from each threat model, SimCat can classify eight different attack types such as PGD-ℓ_2, PGD-ℓ_∞, CW-ℓ_2, PPGD, LPA, StAdv, ReColor, and JPEG-ℓ_∞, with over 40% accuracy. On STL10 data, we apply SimCat as a defense against poisoning attacks, such as BP, CP, FC, CLBD, HTBD, halving the success rate while using only twenty total poisons for training. We find that the detectors generalize well to unseen threat models. Lastly, we investigate the performance of our detection method under adaptive attacks and further boost its robustness against such attacks via adversarial training.*

1. Introduction

Deep learning has been applied to many applications with great success, though a major roadblock to safe deployment of deep models in real world settings is their susceptibility to adversarial attacks. Targeting a model at the time of inference is known as an evasion attack [33], where imperceptible perturbations are made to an input to craft an *adversarial example* that is misclassified by the model. Models can also be attacked during training via poisoning [11], where a small number of adversarial examples are inserted to a training set, so that after training, targeted test samples are misclassified.

To mitigate these vulnerabilities, many defenses have been introduced [22, 34, 9, 19]. However, a number of practical challenges remain. First, most defenses only harden a model against a specific narrow threat model [16]. Thus, an attacker can easily evade the defense by using a different attack. Second, novel attacks are introduced frequently, in a brisk game of cat and mouse, in which the attacker generally has the upper hand [2]. When we combine the rapid development of attacks with the high computational cost of retraining a model to be robust against even a single threat, maintaining the robustness of a model to all threats using typical defenses becomes intractable.

Certain detection based defenses [3] do not require fundamentally changing how the model is trained, which allows for easy application of the detector without disrupting the existing machine learning pipeline. However, many detection based systems require large amounts of data. This is problematic in practice, as an adversary may employ an attack that the defender has never seen before, making it difficult for the defender to have trained the detector to work against it. Even if the defender obtains some examples of the novel attack, the training size may not be sufficient to expand the detector's ability to include the novel attack.

The advantage of a detector that is both broad in scope and inexpensive in training requirements is clear. In this paper, we propose a highly sample efficient detector that also generalizes well to unforeseen attacks. Further, we extend our model to classify adversarial attacks to their respective threat models, similar to [23]. The classification allows for additional defenses to be employed, specific to the attack encountered. Our model is based on pretrained self-supervised encoders, which are capable of efficiently ex-

tracting the semantic content of images, as evident by recent successes in the self-supervised domain that have closed the gap with supervised models.

Specifically, we use a SimCLR encoder pretrained on ImageNet, because we observe that distance in the embedding space of this encoder correlates strongly with human perception (Figure 1). Furthermore, the distance between the SimCLR representations of a clean image and its adversarially perturbed counterpart is similar across various threat models, making the SimCLR distance a strong candidate to proxy the true perceptual threat model, which encapsulates all imperceptible adversarial attacks [19]. While LPIPS [37] has also been shown to be a strong proxy for true perceptual distance, the dimensionality of its feature vectors is massive in comparison to that of SimCLR representations. Our method is called **SimCat**, as it uses a SimCLR encoder to linearly catch adversarial attacks and categorize them to their respective threat models.

By utilizing the highly informative low dimensional embedding space of self-supervised encoders, we find that even a linear model trained on these representations can effectively detect and classify adversarial attacks of a wide variety of types. Our experiments over many threat models and multiple datasets show that, for both evasion and poisoning attacks, SimCat greatly outperforms baseline models. While the method is extremely simple, we argue that its simplicity is essential for the efficiency of the method both in terms of time and sample complexities. By freezing the encoder, SimCat's optimization is *convex* and thus its global optimum can be found effectively. Moreover, SimCat has low model complexity due to the small dimensionality of SimCLR representations, allowing for highly efficient training that also generalizes well. These properties lead to an impressive empirical performance of SimCat in detection and classification of various types of adversarial examples using as few as 5 training samples per class.

For example, on ImageNet, using only 5 samples per threat model, SimCat's detection accuracy is 68.5%, improving the baseline method by more than 6.4%. Using the same setup, SimCat's classification accuracy of 8 adversarial attacks including PGD-ℓ_2, PGD-ℓ_∞, CW-ℓ_2 [4], PPGD [19], LPA [19], StAdv [35], ReColor [18], and JPEG-ℓ_∞[16] is 27.1%, improving the baseline performance by 7.7%. Using 25 samples per class, SimCat's gains over baseline method grows to 7.4% and 11.8% for detection and classification problems, respectively.

Interestingly, SimCat can be used to detect and classify various types of poisoning attacks as well, which we then apply as an efficient poison defense. We consider five types of poisonings including bullseye polytope (BP) and convex polytope (CP) [1, 38]. Using two SimCat detectors trained with 10 BP and 10 CP poisons respectively, we construct an ensemble detector to remove any sample that is flagged as poison by either of the individual detectors. The ensemble reduces poison success rate over five types of poisoning attacks from 21.8% to 9.7%. Notably, the SimCat poisoning defense only reduces clean accuracy by 1%.

Lastly, we develop an adaptive attack that creates adversarial examples to evade SimCat detection. We then design an adversarial training procedure with momentum updates and data augmentation to improve robustness of SimCat against adaptive attacks. On ImageNet, the SimCat detector adversarially trained using 25 samples per threat model achieves 71.7% robustness to a PGD-ℓ_2 ($\epsilon = 2.0$) adaptive attack, a 32% improvement over vanilla SimCat. Moreover, the adversarially trained SimCat improves the clean accuracy as well, from 73.2% to 73.6%.

In summary, we make the following contributions:

- We identify that pre-trained SimCLR embeddings contain valuable information regarding perceptibility of adversarial perturbations. Using this intuition, we develop a sample efficient method for detection and classification of adversarial attacks called SimCat.
- We demonstrate the effectiveness of SimCat in detection and classification of various types of adversarial examples in test time (evasion attacks) and training time (poisoning attacks). SimCat leads to impressive empirical results on the ImageNet scale using as few as five training samples per class.
- We study adaptive attacks against SimCat and develop an adversarial training procedure that dramatically increases its robustness to adaptive attacks while improving its clean accuracy.

2. Prior Works

2.1. Adversarial Attacks and Defenses

Given an input $x \in X$ with label $y \in Y$ and a classifier $\mathbf{f} : X \to Y$, an adversarial attack $\hat{x}$ satisfies $\mathbf{f}(\hat{x}) \neq y, d(\hat{x} - x) \leq \epsilon$, for some generally small bound ϵ. Here, $d(\cdot, \cdot)$ is a distance metric that defines the threat model (i.e. the space of allowable perturbations to the input to craft the adversarial attack). Threat models using ℓ_2 and ℓ_∞ distance are well studied [5, 22], though attacks that apply spatial transformations, recoloring, frequency domain perturbations [35, 18, 16] are also effective. Ideally, a defense would ensure that any two images that are imperceptible to a human are classified in the same way. This motivates the neural perceptual threat model of [19], which utilizes LPIPS distance as a proxy for the true perceptual distance.

Defenses against adversarial attacks either focus on robust prediction or detection. Adversarial training [22] is the most common method for robust prediction. It operates by crafting and training on adversarial examples, with the

ground truth label. While it improves robustness for a specific threat model, the gains do not extend to others. Provable defenses based on smoothing have also been proposed, though they pertain to restricted threat models [29, 9].

Attacks can also be made during training, known as data poisoning [11], where a training set is corrupted so that the trained model misclassifies certain target samples. Clean label poisoning attacks are particularly dangerous and covert, as poisons have the correct label, so the accuracy of the model after training remains high. Two of the strongest clean label poisonings attacks are Convex Polytope (CP) [38] and Bullseye Polytope (BP) [1], which work by making imperceptible changes to a set of baseline images from an intended class so that the features of the baseline images surround a target image, who at test time is then classified to the class of the baseline images. Naturally, robust prediction based defenses can not be applied to data poisoning.

A number of supervised detection methods have been proposed, based on deep network activations [24, 6], statistical tests [13, 28], local intrinsic dimensionality [21], to name a few. Unsupervised methods based on feature squeezing [36], generative models [32], nearest neighbor search, KL divergence [25], among others, have also been suggested. For a comprehensive review, we refer readers to [3]. Generally, unsupervised methods can be costly to configure and sensitive to noise, while supervised methods require lots of data and often fail to generalize to unseen threats. Many detectors have also been shown to be vulnerable [5]. To the best of our knowledge, there is no detection system that uses as few samples as SimCat.

Classifying adversarial examples to their respective threat models has been explored in [23, 20]. This classification can allow for more specific defenses to be applied off the shelf when appropriate, and also give the defender insight about the attacker.

2.2. Self-Supervised Encoders

Recent work has seen self-supervised models advance rapidly and in multiple domains [8, 27]. We focus on SimCLR [7], which is trained using contrastive learning.

Contrastive Learning is a simple yet powerful self-supervised framework for representation learning that has made large strides in closing the gap with supervised learning. The contrastive loss seeks to maximize similarity between representations of two views of an input, and minimize similarity to views of other samples. SimCLR uses this simple framework, along with a multi-step data augmentation pipeline for generating different views of the same image, to learn very informative visual representations of images. Specifically, SimCLR indirectly applies the contrastive loss on the representations by way of a shallow MLP projection network appended to the encoder during training, and discarded afterwards. While training self-supervised models can be computationally expensive, in our experiments, we use a *fixed* SimCLR encoder *pre-trained* on ImageNet, available openly from [10].

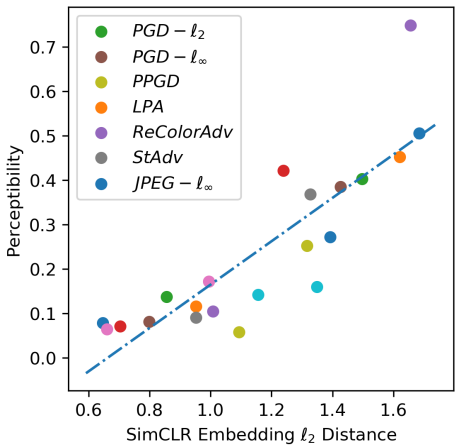

Figure 1. Perceptibility of adversarial attacks relative to the ℓ_2 distance between the original and perturbed image in the SimCLR embedding space. Each point in the scatter plot refers to the average distance between adversarial examples within a single threat model under one of three bounds. Correlation is $r = 0.854$.

A few works have looked into adversarially robust contrastive learning [12, 14, 17, 15], though our work differs in that our encoder is fixed and applied to detection and classification of adversarial attacks. Most similar to our work is [31], where SimCLR embeddings are used for anomaly detection. To our knowledge, our work is the first to identify the SimCLR embedding space as one in which adversarial examples and clean images seem to be linearly separable.

3. SimCLR distance as proxy for Perceptibility

In this section, we outline the motivation behind using SimCLR as the self-supervised encoder for SimCat. We make use of the data from the human perceptual study in [4]. The data consists of seven threat models, spanning perceptual, ℓ_p, spatial, recoloring, and compression attacks, under three levels of bound on the applied perturbations. Humans were then used to evaluate how perceptible the perturbations were. This was done by presenting a clean and adversarially perturbed sample side by side for two seconds, then having the participant choose whether they thought the images were the same or different. This gives a notion of perceptibly, measured as the ratio of humans who felt the attacked image looked distinct from the original.

In figure 1, the mean perceptibility over each threat model and attack bound pair is plotted, against the mean ℓ_2 distance between SimCLR representations of the clean and attacked image. We observe a strong correlation, with Pearson's $r = 0.854$. The correlation is reduced by the high perceptibility of the large coloring attacks. This can

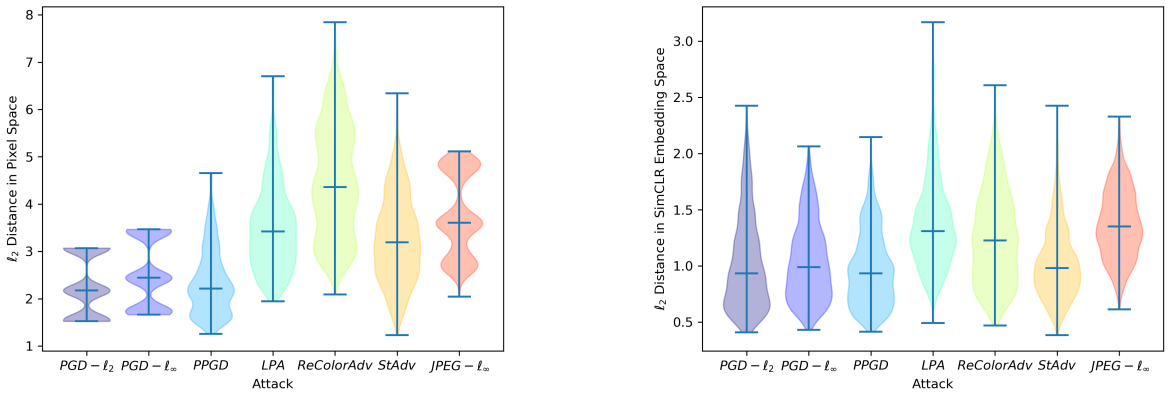

Figure 2. The distributions of ℓ_2 distances between adversarial examples and clean images in pixel space (left) and SimCLR embedding space (right). Note the different scales; the distributions across threat models are much more uniform when using SimCLR embeddings.

be explained by the fact that SimCLR is trained to be less sensitive to color shifts, as color jitter is an important augmentation employed in the SimCLR pipeline. Removing the coloring attacks, the correlation improves to $r = 0.892$.

Furthermore, SimCLR distance scales similarly for diverse attack types. We observe this in figure 2, where some non-ℓ_p attacks require much higher bounds on ℓ_2 distance to be encapsulated. On the other hand, SimCLR distance smoothly distributes the attacks of various types and level. This makes the SimCLR distance a strong proxy for the perceptual threat model, suggesting that it could be useful in adversarial training against unseen threat models, though we leave this to a future work.

We note that the correlation found for ℓ_2 distance in image space and LPIPS are both comparable to the correlation for the SimCLR distance [19]. A key advantage of the SimCLR distance over LPIPS is the low dimensionality of its embeddings. While SimCLR only uses a 2048 dimensional representation vector for each input, LPIPS concatenates flattened feature activations from many layers in a deep network to compute distance, which can lead to a blow up in the size of the representation vector, due to the ever increasing depths and widths of modern deep networks (e.g. for LPIPS evaluated on AlexNet, the representation vector has length upwards of 500,000).

4. Methods

4.1. General SimCat Framework

In this section, we describe our proposed methods. We use a self-supervised encoder $\phi : \mathbb{R}^D \to \mathbb{R}^d$. Importantly, the self-supervised encoder does not need to be trained or finetuned on the data we wish to apply SimCat to. In our experiments, we use a SimCLR encoder with a ResNet50 backbone pretrained on ImageNet to map inputs into a $d = 2048$ dimensional embedding space. Interestingly, the same encoder can be applied effectively to images of varying size (e.g. SVHN (32), STl10 (96), and ImageNet (224)). We apply a linear transformation on the extracted representations to obtain logits.

For detection, we call our model SimCatch, and denote it by $\mathbf{d}_{\phi,\boldsymbol{\omega}}$, where $\boldsymbol{\omega}$ contains all the $d+1$ trainable parameters, consisting of vector weights $\mathbf{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$. The SimCatch detector maps $\mathbf{d}_{\phi,\boldsymbol{\omega}} : \mathcal{X} \cup \hat{\mathcal{X}} \to \mathcal{Y}$, where $\mathcal{X}$ is the space of all natural images, $\hat{\mathcal{X}}$ is the space of all imperceptible adversarial images, and $\mathcal{Y} = \{0, 1\}$ is the space of ground truth binary labels, with 1 denoting an adversarial example. Since it is intractable to capture the entire space of natural and adversarial examples, we estimate $\mathcal{X} \cup \hat{\mathcal{X}}$ with the dataset $D = \bigcup_{i=1}^{N} \{(\mathbf{x_i}, 0), (\hat{\mathbf{x}}_\mathbf{i}, 1)\}$, where $\hat{\mathbf{x}}_i$ is an adversarial example obtained by attacking $\mathbf{x}$. An attack specific detector is obtained by restraining the threat model of adversarial examples included in D, and an attack agnostic detector seeks to approximate the space of all adversarial examples by sampling from multiple diverse threat models. The output of SimCatch on an input image $\mathbf{x}$ is

$$\text{SimCatch}(\mathbf{x}) := \mathbf{d}_{\phi,\boldsymbol{\omega}}(\mathbf{x}) = sgn(\mathbf{w}^\mathbf{T}\boldsymbol{\phi}(\mathbf{x}) + b) \qquad (1)$$

where *sgn* is the sign function. Note that D does not need to consist of clean and attacked pairs; it can also be two unrelated sets of clean and attacked examples. We hypothesize that training on clean and correspondingly attacked pairs will lead to a more precise decision boundary, but we find in practice that using arbitrary clean samples also suffices.

In classification over k threat models, the vector weights $\mathbf{w}$ are replaced with a matrix $\mathbf{W} \in \mathbb{R}^{k \times d}$.The bias b also now becomes a d dimensional vector $\mathbf{b}$. We refer to this model as SimClass with learnable parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}\}$ and denote it as $\mathbf{g}_{\phi,\boldsymbol{\theta}} : \bigcup_{i=1}^{k} \hat{\mathcal{X}}_{d_i} \to [k]$, where $\hat{\mathcal{X}}_{d_i}$ is the space of adversarially perturbed images under a threat model defined by distance metric d_i. The training set $D =$

$\bigcup_{i=1}^{k} \bigcup_{j=1}^{N} \{(\hat{x}_j^i, i)\}$ consists of N adversarially perturbed examples from each of the k threat models. The output of SimCat used as classifier on input x is then

$$\text{SimClass}(\mathbf{x}) := \mathbf{g}_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathbf{x}) = \arg\max_{i \in [k]} (\mathbf{W}\boldsymbol{\phi}(\mathbf{x}) + \mathbf{b})_i \quad (2)$$

Both SimCat models are trained with a cross entropy loss and ℓ_2 regularization. Without loss of generality, we present the optimization formulation for SimCatch below.

$$\min_{\boldsymbol{\omega}} \sum_{(x,y) \in D} \mathcal{L}_{ce}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\mathbf{x}), y) + \lambda \|\boldsymbol{\omega}\|^2 \quad (3)$$

Importantly, in training, the self-supervised encoder $\boldsymbol{\phi}$ is *fixed*. Thus, the number of learnable parameters scales linearly with the number of output classes and dimensionality of the embedding space of $\boldsymbol{\phi}$. Moreover, the optimization is now *convex*. Due to the low dimensionality of SimCLR's encoder and the convex nature of SimCat's optimization, the global optimum can be found efficiently, in both time and sample complexity. In our experiments, we set the regularization constant $\lambda = 1$, and use L-BGFS to obtain optimal parameters for SimCat's regularized logistic regression.

4.2. SimCat Variants

There are many additional modifications that can be made to SimCat to further improve its performance. The majority of the experiments do not use any variants, but in some cases we include the following:

- **Data Augmentation** is a common technique to improve generalizability of deep models. Naturally, data augmentation is very useful when there is limited data available. We utilize data augmentation to balance the dataset during adversarial training (algorithm 1). Our experiments with data augmentation show improvements in extremely low data settings, though only modest improvement in other cases.
- **Ensembling** can produce improved models by way of combining the outputs of multiple independently trained models. We ensemble detectors for specific poisoning types to improve the performance of our SimCatch based defense (table 3).

5. Detection and Classification

5.1. Evasion Attacks

5.1.1 Experimental Set up

We evaluate SimCat's detection and classification capabilities of evasion attacks from two datasets. We compare SimCat's performance to an analagous model that fits a linear layer atop fixed ResNet50 features learned via *supervised* ImageNet pretraining. The baseline differs with SimCat only in that it uses embeddings in a feature space learned with label supervision, highlighting how self-supervised features may better capture distinguishing nuances between the distributions of natural and adversarial images. We present results for a second baseline that additionally finetunes the pretrained ResNet50 in the supplemental material.

For SVHN [26], a dataset of street view house numbers with smaller images (32×32), we perform PGD ℓ_∞ and ℓ_2 attacks, with budgets of $\epsilon = 8/255$ and $\epsilon = 1.0$ respectively. We train a detector for each PGD attack, and a classifier to distinguish between the two ℓ_p threats.

For ImageNet, we use the perceptual study data introduced in Section 3. Specifically, we take the attacks of the 'large' bound, which have a budget of $\epsilon = 8/255$ for the PGD-ℓ_∞ attack. The budget for each attack can be found in the original paper. Additionally, we perform Carlini-Wagner-ℓ_2 attacks on 200 other clean images. We train a *general* detector on all eight attack types, and also a classifier to distinguish between the eight attack types.

For each adversarial sample, we also have the original clean image. The samples are divided so that the pairs remain in the same set, ensuring that we never have a test image that is either the clean or adversarially perturbed version of a training image. Thus, the total training set size for a detection trial is equal to the number of training samples per attack times 2 times number of attacks, and times k for k-way classification.

In table 5.1.1, we present results averaged over ten trials, so to account for variability introduced by sampling such a small fraction of the data to train each model.

5.1.2 Results

SimCat outperforms the baseline across the board especially for SVHN, reaching increases in accuracy of as high as 21.0%. The efficiency of SimCat is highlighted in SVHN PGD-ℓ_∞ detection, where fitting a detector to just two adversarial examples yields 77.3% accuracy. SimCat's largest gains over the baseline comes in classification tasks (3), indicating that self-supervised features are more sensitive to the distinguishing characteristics of specific attack types.

5.2. Poisoning Attacks

5.2.1 Experimental Setup

We test SimCat on five poisoning attacks, including Bullseye Polytope (BP), Convex Polytope (CP), Feature Collision (FC), Clean-label Backdoor (CLBD), and Hidden-trigger Backdoor (HTBD). Poisons are generated using the white-box transfer learning set up as described in [30], where the attacker seeks to poison a fine-tuning set for transfer learning. This setting is generous to the attacker, as they only need to poison a linear layer appended to a fixed feature encoder, that they also have access to. Furthermore,

SVHN		Training Samples per Attack				
Task	Attacks	2	5	10	25	50
Detection	PGD-ℓ_2	63.3(+8.8)	71.9(+11.5)	75.0(+11.9)	81.7(+13.1)	85.7(+12.0)
Detection	PGD-ℓ_∞	77.3(+15.0)	82.5(+13.9)	88.5(+14.0)	92.4 (+9.9)	94.2(+8.4)
Classification	PGD ℓ_2, PGD ℓ_∞	60.6(+8.9)	64.1(+12.8)	70.9(+16.3)	77.1(+21.0)	81.5(+19.6)

IMAGENET		Training Samples per Attack				
Task	Attacks	5	10	25	50	100
Detection	PGD-ℓ_2, PGD-ℓ_∞, PPGD, LPA, JPEG-ℓ_∞, StAdv, ReColor, CW-ℓ_2	68.5(+6.4)	71.5(+7.4)	74.3(+7.4)	76.5(+5.3)	79.2(+4.4)
Classification		27.1 (+7.7)	32.8(+10.2)	40.7(+11.8)	48.9(+12.8)	58.1(+15.3)

Table 1. Performance of SimCat for detection and classification using few training samples on SVHN (top) and ImageNet (bottom). In parenthesis, we denote the improvement gained by using SimCat compared to a baseline using supervised embeddings. For ImageNet, the detector is trained and evaluated over all eight attack types at once, and classification is done over all eight attack types.

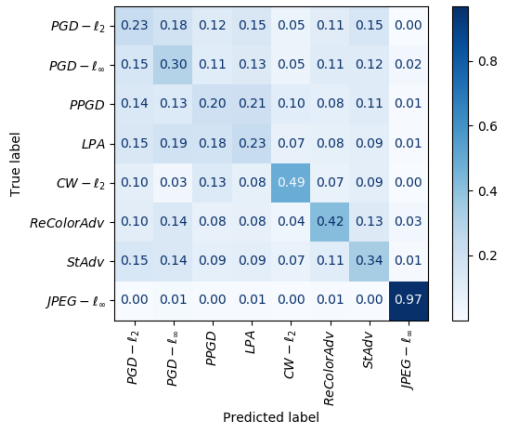

Figure 3. SimCat classification accuracy over eight diverse attack types. The classifier is fit on just 25 samples per class. Overall classification accuracy is 40.7%, an 11.8% increase over baseline.

Detection		Classification	
Task	Accuracy	Task	Accuracy
BP	85.3	Backdoor vs. Triggerless	68.9, 78.4*
CP	84.1		
General	64.5, 70.5*	5-way	52.4

Table 2. Results for SimCat detection and classification of poisoning attacks on STL10 using ten samples per poisoning. Asterisk indicates removing FC poisons.

Attack	Standard		SimCat		Ens. SimCat	
Type	Acc	PSR	Acc	PSR	Acc	PSR
BP	86	82	86	54	85	41.3
CP	86	24	86	11.3	85	4
FC	87	0	86	1.3	85	2
CLBD	87	0	86	0.7	85	0
HTBD	86	2	86	2	85	1.3
Avg	86	21.8	86	13.9	85	**9.7**

Table 3. Poison defense via SimCat detection. PSR is poison success rate. SimCat model is trained on CP and BP jointly. Ens. SimCat trains a CP detector and a BP detector separately, then filters any samples that are detected as poison by either detector. Both defenses only use *ten* samples of CP and BP poisons each.

the training set sizes are small. Specifically, the finetuning set only uses 2500 images, and the attacker can insert 1% (25) additional samples. We use the STL10 dataset, as an intermediary between SVHN and ImageNet. As an additional challenge, we use the SimCLR encoder as the fixed feature encoder – this means that the poisoning attacks directly cause collisions with clean samples in the same space we wish to use to distinguish them. We test how well clean target samples can be distinguished from poisons that (by design) would be in close proximity to the targets in SimCLR space. We also test SimCat as a poison defense.

5.2.2 Results

SimCat again shows strong detection and classification accuracy with high sample efficiency (2), particularly for BP and CP poisons, which happen to be the strongest. SimCat struggles with FC poisons, most likely since FC poisons are designed to directly collide with target representations, while BP and CP poisons surround a target instead. When excluding FC poisons, general detection rises to 70.5%.

We then apply SimCat detectors as a poison defense. We use BP and CP poisons to train the detectors, since those poisons are most lethal. Using ten samples each, we train an attack agnostic detector, and two separate detectors specific to each threat model, which are used as an ensemble detector, that only admits samples deemed clean by both detectors. Table 3 shows that both the general detector and the ensemble detector are effective, with the ensemble detector reducing poison success rate from 21.8% to 9.7%,

Trained on:	# of Samples	PGD-ℓ_2	PGD-ℓ_∞	PPGD	LPA	CW-ℓ_2	ReColor	StAdv	JPEG-ℓ_∞	Avg.
Single Attack	100	68.8	67.9	68.5	69.2	62.2	64.9	65.6	50.2	64.7
Union of	5	66.4	70.7	65.6	73.4	51.5	71.4	63.3	69.9	66.5
Attacks	20	71.1	76.6	69.0	80.1	51.1	74.6	66.1	79.4	71.1

Table 4. Generalization of SimCat detectors to unseen threat models. First row shows accuracy of detector trained on a single attack evaluated on all other attacks. Other rows contain accuracy of a SimCat detector trained on the union of all other attacks. The second column indicates the number of samples per threat model used in training.

while also maintaining high clean accuracy.

6. Generalization to Unseen Models

Generalization of defenses to unseen attacks is of utmost importance because of the constant development of novel threats. In table 4, we see that SimCat generalizes surprisingly well given its simplicity. Even when trained only on a single threat model, some of the SimCat detectors achieve close to 70% detection accuracy on unseen attacks. The generalization of detectors trained on the union of attacks is also impressive, particularly given the sample efficiency. We observe that the detector trained on the union of attacks with just five samples per threat model (35 total) exceeds the average accuracy on unseen attacks achieved by detectors trained on a single threat model with 100 training samples.

In figure 4, we get a closer look at how each threat model generalize to others. The detectors trained on the perceptual attacks (PPGD, LPA) generalize the best to unseen threats. This invokes our motivating observation that the SimCLR embedding space seems to contain information pertinent to perceptibility. Understanding how human and machine perceptions differ is at the heart of many vision tasks, including adversarial robustness, and we encourage future work to further investigate the semantic meaning extracted in SimCLR and other self-supervised models.

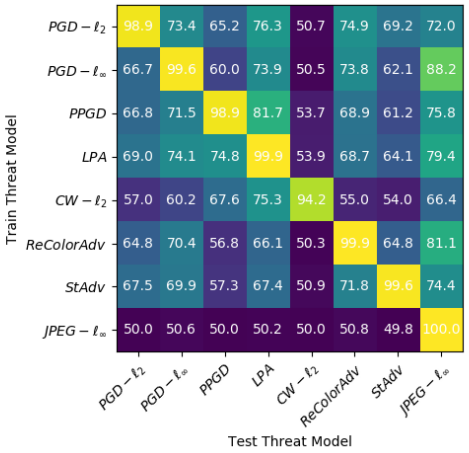

Figure 4. Generalizability of SimCat detectors to unseen threat models. Each detector is trained on 100 samples from a single threat model, and evaluated on all other models.

7. Hardening SimCat to an Adaptive Attack

In this section, we investigate the robustness of SimCat to an adaptive attack. An adaptive attack is an attack that is specifically crafted based on knowledge of a model's defense. By investigating adaptive attacks for SimCat, we can identify limitations of our model, and work towards mitigating them (i.e. via adversarial training) before the vulnerability is exposed and exploited by an adversary.

7.1. Attack Formulation

We consider a white box attack setting, where the attacker has knowledge of the base classifier *and* the SimCat detector. The ultimate goal is to cause a misclassification in the base classifier, but the adversary must first evade the SimCat detector. Denoting the base classifier as $\mathbf{f}$, the detector as $\mathbf{d}$, and an input-label pair as (x, y), we formulate the adaptive adversarial attack problem as the following.

$$\boldsymbol{\delta} = \underset{\boldsymbol{\delta}, \|\boldsymbol{\delta}\| \leq \epsilon}{\arg\max}\, \mathcal{L}(\mathbf{f}(\hat{\mathbf{x}} + \boldsymbol{\delta}), y) + \mathcal{L}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\hat{\mathbf{x}} + \boldsymbol{\delta}), 1) \quad (4)$$

For both terms, $\mathcal{L}$ is the cross entropy loss. The detector outputs 1 for adversarial examples, so the adaptive attack seeks to flip this label by maximizing the loss incurred by it. We solve the above optimization with projected gradient descent, and find that the adaptive attack is somewhat effective against an undefended SimCat ImageNet detector, reducing accuracy by 30% (table 5).

7.2. Adversarial Training

We employ adversarial training (AT) to improve the robustness of SimCat to the adaptive attacks described in the previous section. Standard AT seeks to harden a network by crafting adversarial examples throughout training, and additionally training the model on the crafted examples with the original label. For SimCat, this amounts to the following min-max optimization, where $\mathbf{d}$ is the SimCat detector with parameters $\boldsymbol{\phi}, \boldsymbol{\omega}$, and $\mathbf{f}$ is the base classifier.

$$\min_{\boldsymbol{\omega}} \max_{\boldsymbol{\delta}; \|\boldsymbol{\delta}\| \leq \epsilon} \mathcal{L}_{ce}(\mathbf{f}(\hat{\mathbf{x}} + \boldsymbol{\delta}), y) + \lambda \mathcal{L}_{ce}(\mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}(\hat{\mathbf{x}} + \boldsymbol{\delta}), 1) \quad (5)$$

SimCat AT is different from standard AT in a few ways. Standard AT usually takes a few steps of finding perturba-

tions to increase the objective, followed by a few steps of updating model parameters to reduce the objective. In SimCat AT, the minimization step is solved to completion after having crafted adaptive adversarial examples, as opposed to only taking a few steps. This can be done efficiently due to the low number of parameters to solve for and the convexity of the minimization problem. A couple other steps are needed for SimCat AT to be effective.

- Momentum updates are used to stabilize training. The importance of momentum updates is clear in figure 5, as the $\beta = \infty$ case (where the SimCat detector is replaced in each epoch with the optimal solution for detecting the current batch of adaptive adversarial examples) yields worse robustness than a standard SimCat.

- Along with the additional adaptive adversarial attacks, the original data is retained in the training set for each iteration, so to mitigate a robustness-accuracy tradeoff. To balance the dataset, an augmented copy of the clean samples is also added to the training set in each epoch.

Algorithm 1 Adversarial Training of SimCat: inputs are the base classifier $\mathbf{f}$ and data $\left(\{(x_i, \hat{x}_i)\}_{i=1}^N\right)$, where x is a clean sample and $\hat{x}$ is x after being adversarially perturbed.

Obtain initial parameters via standard SimCat:
$\boldsymbol{\omega} \leftarrow \textsc{FitSimCat}\left(\{(x_i, \hat{x}_i)\}_{i=1}^N\right)$
Augment clean data to obtain second copy:
$\{\tilde{x}_i\}_{i=1}^N \leftarrow \textsc{Augment}\left(\{x_i\}_{i=1}^N\right)$
for $t = 1, \ldots,$ # of epochs **do**
  Craft perturbations for adversarial examples to evade both detector and base classifier:
  $\{\delta_{i,t}\}_{i=1}^N = \textsc{AdaptivePGD}\left(\{\hat{x}_i\}_{i=1}^N, \mathbf{f}, \mathbf{d}_{\boldsymbol{\phi},\boldsymbol{\omega}}\right)$
  Solve SimCat with expanded dataset:
  $\boldsymbol{\omega}_t \leftarrow \textsc{FitSimCat}\left(\{(x_i, \hat{x}_i), (\tilde{x}_i, \hat{x}_i + \delta_{i,t})\}_{i=1}^N\right)$
  Apply momentum update to SimCat parameters:
  $\boldsymbol{\omega} \leftarrow (\boldsymbol{\omega} + \beta\boldsymbol{\omega}_t)/(1+\beta)$
end for

7.3. Results

While the adaptive adversarial attack is somewhat effective (reducing attack detection accuracy by 34%), the SimCat AT algorithm completely recovers robust accuracy, while also improving overall accuracy. Table 5 shows the effect of AT and augmentation, which in tandem become a very strong defense. Furthermore, the entire adversarial training procedure takes only about fifteen minutes, and all SimCat AT model results presented used only 25 training samples per attack. Thus, the SimCat framework lends itself to increased robustness via algorithm 1, without compromising training and data efficiency.

Model	Accuracy	Robustness
SimCat	73.21	39.25
SimCat+Aug	74.87	37.40
SimCat+AT	74.23	67.95
SimCat+AT+Aug	**73.55**	**71.70**

Table 5. Ablation study on SimCat AT. Accuracy refers to attack agnostic ImageNet detection from Section 5.1. Robustness is measured as the percent of test adversarial samples that can be adaptively attacked with PGD-ℓ_2, $\epsilon = 2.0$ to be misdetected as clean. AT is done for 20 epochs. Aug refers to augmenting both the original clean and adversarial samples - this is distinct from AT+Aug, where only the clean samples are augmented to balance out the addition of adaptive adversarial attacks to the SimCat training set.

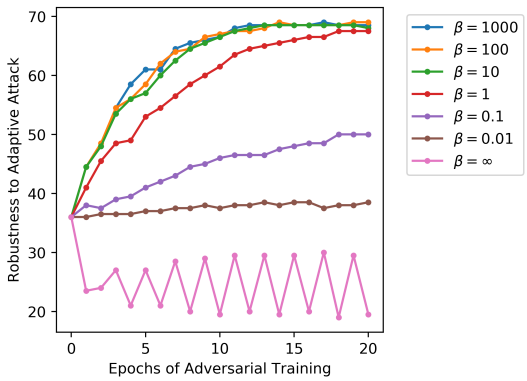

Figure 5. SimCat robustness to adaptive PGD-ℓ_2 over epochs of adversarial training with varied values of the hyperparameter β, which controls the momentum updates. Higher values of β lead to more emphasis on the linear classifiers solved in later epochs. Adversarial training is unstable without momentum ($\beta = \infty$).

8. Conclusion

In this paper, we introduced SimCat, a sample efficient method for detection and classification of adversarial attacks. SimCat uses a linear model over embeddings of a pre-trained self-supervised model, SimCLR. SimCat is successful in detecting and classifying various types of adversarial attacks ranging from ℓ_p and non-ℓ_p evasion attacks to poisining attacks, likely because pre-trained SimCLR embeddings can be used to uniformly quantify perceptibility of various types of adversarial perturbations. Over various experiments on SVHN, ImageNet, and STL10 datasets, we demonstrate the effectiveness of SimCat using as few as two training samples per class. We have also studied adaptive attacks against SimCat and developed an adversarial training procedure that dramatically increases its robustness against such attacks while improving its clean accuracy.

9. Acknowledgements

This project was supported in part by NSF CAREER AWARD 1942230, HR00112090132, HR001119S0026 and ONR GRANT13370299.

References

[1] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability, 2021. 2, 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018. 1

[3] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, and D. Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020. 1, 3

[4] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017. 2, 3

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. 2, 3

[6] Fabio Carrara, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato. Adversarial examples detection in features distance spaces. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 313–327, Cham, 2019. Springer International Publishing. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 3

[8] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners, 2020. 3

[9] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019. 1, 3

[10] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020. 3

[11] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses, 2020. 1, 3

[12] Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2021. 3

[13] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples, 2017. 3

[14] Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17081–17093. Curran Associates, Inc., 2020. 3

[15] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16199–16210. Curran Associates, Inc., 2020. 3

[16] Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019. 1, 2

[17] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2983–2994. Curran Associates, Inc., 2020. 3

[18] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In *NeurIPS*, 2019. 2

[19] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on Learning Representations*, 2021. 1, 2, 4

[20] Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, D. Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *ArXiv*, abs/2012.01654, 2020. 3

[21] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018. 3

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2

[23] Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Perturbation type categorization for multiple ℓ_p bounded adversarial robustness, 2021. 1, 3

[24] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *Proceedings of 5th International Conference on Learning Representations (ICLR)*, 2017. 3

[25] D. J. Miller, Y. Wang, and G. Kesidis. Anomaly detection of attacks (ada) on dnn classifiers at test time. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. 3

[26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 5

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 3

[28] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR, 09–15 Jun 2019. 3

[29] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. Denoised smoothing: A provable defense for

pretrained classifiers. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21945–21957. Curran Associates, Inc., 2020. 3

[30] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks, 2020. 5

[31] Vikash Sehwag, Mung Chiang, and Prateek Mittal. {SSD}: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021. 3

[32] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. 10 2017. 3

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1

[34] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. 1

[35] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. 2

[36] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *CoRR*, abs/1704.01155, 2017. 3

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2

[38] Chen Zhu, W. Ronny Huang, Ali Shafahi, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets, 2019. 2, 3