Adversarial Defense based on Structure-to-Signal Autoencoders

Joachim Folz* Sebastian Palacio* Joern Hees Andreas Dengel
German Research Center for Artificial Intelligence (DFKI)
TU Kaiserslautern
first.last@dfki.de

Abstract

Adversarial attacks have exposed the intricacies of the complex loss surfaces approximated by neural networks. In this paper, we present a defense strategy against gradient-based attacks, on the premise that input gradients need to expose information about the semantic manifold for attacks to be successful. We propose an architecture based on compressive autoencoders (AEs) with a two-stage training scheme, creating not only an architectural bottleneck *but also a* representational bottleneck*. We show that the proposed mechanism yields robust results against a collection of gradient-based attacks under challenging white-box conditions. This defense is attack-agnostic and can, therefore, be used for arbitrary pre-trained models, while not compromising the original performance. These claims are supported by experiments conducted with state-of-the-art image classifiers (ResNet50 and Inception v3), on the full ImageNet validation set. Experiments, including counterfactual analysis, empirically show that the robustness stems from a shift in the distribution of input gradients, which mitigates the effect of tested adversarial attack methods. Gradients propagated through the proposed AEs represent less semantic information and instead point to low-level structural features.*

1. Introduction

We nowadays see an increasing adoption of deep learning techniques in production systems, including safety-relevant ones (as exemplified in [28]). Hence, it is not surprising that the discovery of adversarial spaces for neural networks [40] has sparked a lot of interest and concern. A growing community focuses on different ways to reach those spaces [12, 25, 7], understand their properties [28, 16, 23], and protect vulnerable models from their malicious nature [29, 11, 21].

Commonly used methods to exploit adversarial spaces rely on input gradients as basis to find perturbations surrounding an otherwise clean sample [12, 25, 7, 21]. Due to the intractability of transformations modeled by neural networks, and the limited amount of change that is allowed for perturbations to be considered adversarial, the gradient of the model w.r.t. a given sample exposes just enough information about input dimensions that correlate highly with the predicted class. With this in mind, defenses against adversarial attacks that rely on gradients have been devised in two fundamental ways: 1) They complement traditional optimization schemes with a two-fold objective that minimizes the overall prediction cost while maximizing the perturbation space around clean images [21, 35, 43]. 2) Gradients are blocked or obfuscated in such a way that attack algorithms can no longer use them to find effective adversarial perturbations [5, 13, 36]. Type 1 methods enjoy mathematical rigor and hence, provide formal guarantees with respect to the kind of perturbations they are robust to. However, these regimes have strong limitations, since assumptions about the nature of the attacks have to be made to formulate the optimization target. Methods of type 2 have been systematically proven ineffective through general methods to circumvent said defense mechanisms [2, 9, 1]. Furthermore, while effective for small-scale problems such as MNIST and CIFAR-10, we found no empirical evidence that these methods scale to larger problems such as ImageNet. In fact, it has been shown that defenses for adversarial attacks tested on small datasets do not scale well when applied to bigger problems [16].

Several emerging defenses for large-scale scenarios are still reliant on the second use of gradients: suppression [19, 47] and blockage [13, 46]. These are instances of *gradient obfuscation* as defined by Athalye *et al*. [2], where instabilities from vanishing or exploding gradients, and stochastic or non-differentiable preprocessing steps are used to defend the model. As demonstrated also in [2], defenses based on obfuscation can be easily circumvented. Hence the aforementioned defense mechanisms are still susceptible to adversarial attacks. In this paper, we propose an alternative defense that affects the information contained in gradients

*Authors contributed equally

by reforming its class-related signal into a structural one. Intuitively, we learn an identity function that encodes low-level features and decodes only the structural parts of the input necessary for classification, dropping everything else. To this end, an autoencoder (AE) is trained to approximate the identity function that preserves only the portion of the original signal that is *useful* for a target classifier. The structural information is preserved by training both encoder and decoder unsupervised, and fine-tuning only the decoder with gradients coming from an pre-trained classifier. This second step induces a *representational bottleneck* in the decoder that complements the architectural compression layer of the AE, aligning reconstructions with the latent classification manifold in a straightforward fashion. By using a function that only encodes structure (the first half of the AE), input gradients are devoid of any class-related information, therefore invalidating the fundamental assumption about gradients that attackers rely on. We show that gradient-based attacks fail to generate perturbations that are adversarial for a pre-trained classifier when gradients are extracted from said fine-tuned AE. Moreover, we show that the ineffectiveness of said attacks is closely related to the kind of input gradient they receive as input, and not because of instabilities *i.e.*, gradient obfuscation. We refer to the final AE architecture as a *Structure-To-Signal* autoencoder (S2SAE). Similarly, we define the term *Structure-to-Signal* network (S2SNet) for the ensemble of an S2SAE and classifier.

1.1. Formal Definitions

To understand the relationship between adversarial attacks, input gradients and adversarial perturbations, we introduce the following notation:

Let $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$ be a continuously differentiable classifier for k classes, and $\tilde{x}_f$ the portion of the signal in the original input x that is effectively being used by f, *i.e.*, $f(x) = f(\tilde{x}_f)$. Typical state-of-the-art classifiers only use a fraction of the available information [27], so for some comparative measure of information content $\mathcal{I}$ (*e.g.*, *normalized mutual information* [37]) it holds that $\mathcal{I}(x, \tilde{x}_f) \ll 1$. Let $\nabla_x f \subseteq \mathbb{R}^n$ refer to the sensitivity of input samples w.r.t. the cost function used for training f, *e.g.*, categorical crossentropy. Let $\mathcal{P} = \{\delta : ||\delta||_p < \epsilon\} \subseteq \mathbb{R}^n$ be the space of all adversarial and non-adversarial perturbations under a given norm $p \in \{0, 1, 2, \infty\}$. Perturbation vectors $\delta \in \mathcal{P}$ are applied to input samples by point-wise addition, creating perturbed samples $\hat{x} = x + \delta \in \mathbb{R}^n$. We also define the sub-space $\mathcal{P}_{x,f} \subseteq \mathcal{P}$ as the set of adversarial perturbations $\delta_{x,f}$ that are reachable from the input gradients $\nabla_x f$. Adversarial perturbations are elements $\delta_{x,f} \in \mathcal{P}_{x,f}$ such that $y = f(x) \neq f(x + \delta_{x,f})$ [1]. A

[1] Adversarial perturbations can also be reached through other domains that do not depend on gradients [26]. We restrict our analysis to gradient-based attacks.

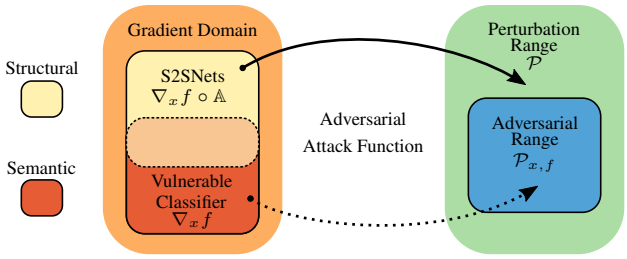

Figure 1: S2SNets shift the domain of an attack function from semantic ($\nabla_x f$) to structural gradients ($\nabla_x f \circ \mathbb{A}$). With our defense (solid line), reachable perturbations fall into $\mathcal{P} - \mathcal{P}_{x,f}$ instead of the adversarial range $\mathcal{P}_{x,f}$ (dotted line).

gradient-based attack (or simply an attack) $\alpha : \nabla_x f \rightarrow \mathcal{P}$ is a function mapping input gradients based on a target classifier to perturbations. An attack is adversarial if the set of reachable perturbations is adversarial. In other words, α is adversarial if $\alpha : \nabla_x f \rightarrow \mathcal{P}_{x,f}$. An S2SAE can therefore be defined as a function $\mathbb{A} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that $\mathbb{A}(x) = \tilde{x}_f$. This way, $\mathbb{A}$ is considered an adversarial defense for f against gradient-based attacks if:

$$\begin{aligned} f(x) &\approx f(\mathbb{A}(x)) \\ \nabla_x f \cap \nabla_x f \circ \mathbb{A} &= \emptyset \end{aligned} \tag{1}$$

Intuitively, these conditions guarantee that $\mathbb{A}$ lets semantically relevant information reach the classifier, while exposing only structure-related input gradients so that an attacker cannot reach the space of adversarial perturbations for f. See Figure 1 for a graphical interpretation of this setup. We show through large-scale empirical experiments that our proposed S2SAE meets these two conditions. Details of the architecture and training of the S2SAE and the S2SNet are discussed in Section 3.1. This architecture has several key advantages over other strategies:

- **No compromise for clean inputs**: the first condition in Equation 1 requires that clean samples are classified the same way regardless of the presence or absence of the defense. Some well-known methods have proven robust against attacks at the expense of an often overlooked drop in clean accuracy [12, 15, 47]. As the transformation done by S2SNets preserves the required signal $\tilde{x}_f$ for classification, using either $\mathbb{A}(x)$ or x has no negative impact on the classifier when clean images are used.

- **Attack agnostic**: S2SNets rely on the semantic manifold of the classifier, and not on the specific artifacts introduced by the attacks themselves. This indicates that S2SNets are robust against a plurality of gradient-based attacks at once.

- ***Post-hoc* implementation**: our proposed defense uses information from pre-trained networks, and therefore can be used to defend models that are already available. Additionally, no fine-tuning or architectural adaptations are required to integrate the defense to an existing classifier. Moreover, no additional considerations need to be made when training a new classifier from scratch.

- **Compatibility with other defenses**: due to the compositional nature of this approach, any additional defense strategies that work with the original classifier can be implemented for the S2SNet ensemble.

We test S2SNets on two high-performing image classifiers (ResNet50 [14] and Inception-v3 [38]) against three gradient-based attacks (FGSM [12], BIM [17] and CW [7]) on the large scale ImageNet [33] dataset. Experiments are conducted for both classifiers under the most challenging white-box threat model where the attacker has access to the entire S2SNet ensemble, the input gradients and output predictions. An evaluation of the effectiveness of S2SNets over regular AEs (*e.g.*, as proposed in [22]) is presented in Section 3.1, empirically showing how S2SNets impose an information bottleneck that is harmless for the classifier but detrimental for attackers (first condition of Equation 1). Furthermore, we extract measures from the gradient spaces $\mathcal{P}_{x,f}$ and $\mathcal{P}_{x,f\circ\mathbb{A}}$ to show that the second condition of Equation 1 also holds. In other words, we show that input gradients from an S2SNet cannot be consistently used by an attacker to generate adversarial perturbations.

2. Related Work

The phenomenon of adversarial attacks has gained momentum since its discovery [40] and has had three main areas of focus. The first area is the one that seeks new and more effective ways of reaching adversarial spaces. In [12], a first comprehensive analysis of the extent of adversarial spaces was explored, proposing a fast method to compute perturbations based on the sign of gradients. An iterative version of this method was later introduced [17] and shown to work significantly better, even when applied to images that were physically printed and digitized again. A prominent alternative to attacks based on gradients succeeded using evolutionary algorithms [26]. Nevertheless, this has not been a practical wide spread method, mostly due to how costly it is to compute. Papernot *et al*. [28] showed how effective adversarial attacks could get with very few assumptions about the attacked model. Finally, more elaborate methods that go beyond a greedy iteration over the gradient space perform different kinds of optimization that maximize misclassification while minimizing the norm of the perturbation [25, 7].

The second area focuses on understanding the properties of adversarial perturbations. The work of Goodfellow *et al*. [12] already pointed at the linear nature of neural networks as the main enabler of adversarial attacks. This went in opposition of what was initially theorized, where the claim was that non-linearities were the main vulnerability. Not only was it possible to perturb natural looking images to look like something completely different, but it was also possible to get models issuing predictions with high confidence using either noise images, or highly artificial patterns [40, 26]. Transferring adversarial perturbations was shown to be possible by crafting attacks on one network and using them to fool a second classifier [20]. However, transferable attacks are limited to the simpler methods as iterative ones tend to exploit specific particularities of each model, and are hence less effective when transferred [16]. As it turns out, not only is adversarial noise transferable between models but it is also possible to transfer a single universal adversarial perturbation to all samples in a dataset to achieve high misclassification rates [23, 30]. Said individual perturbations can even be applied to physical objects and bias a model towards a specific class [4]. Further fundamental work has focused on the inescapable nature of adversarial perturbations [34], the influence of optimizers, and the topology of decision boundaries they converge to [41, 10, 24].

The third area of research has focused on how networks can be protected against such attacks. Strategies include changing the optimization objective to account for possible adversarial spaces [21, 35, 15], detection [11], dataset augmentation that includes adversarial examples [12, 43, 6], suppressing perturbations [29, 36, 19, 22, 8, 47] or obfuscating the gradients to prevent attackers from estimating an effective perturbation [5, 13, 46, 44, 31].

While some of these empirical defenses have been shown to work under simple test conditions, it has been demonstrated that reliance on gradient obfuscation is not effective, and that said methods can be easily circumvented with some slight adjustments of the threat model (*i.e.*, the attacking conditions) [2, 1].

In this work, we build upon the idea of using AEs as a compressed representation of the input signal. AEs have been proposed multiple times in the past with different goals in mind: they have been used to limit the dimensionality of the input space and project adversarially perturbed samples back to the space of clean inputs [22, 36]. These defenses rely heavily on the architectural bottleneck of the AE and are limited to an empirical evaluation of the model's accuracy under perturbations with bounded norms for toy datasets. Moreover, instead of operating in the original input space, AEs have been fine-tuned to approximate the residual of adversarial samples (*i.e.*, the perturbation) so that it can be explicitly subtracted before being passed on to a classifier [19]. Most recently, AEs were trained on a dataset augmented with adversarial samples from a classi-

fier which is also trained concurrently [18]. These methods constitute either examples of gradient obfuscation (and therefore, are known to be easy to circumvent [2]) or a less flexible, model- and attack-dependent defense. We propose a fine-tuning stage for the decoder that introduces a *representational bottleneck* [39] to complement the existing *architectural bottleneck* of the AE. Input reconstructions show how this data-driven defense changes the information retained by the AE (and consequently, the input gradient), preserving the input signal that closely relates to the semantic manifold of the classifier. Furthermore we evaluate our claims by conducting a large-scale evaluation on the challenging, full ImageNet validation set, as well as counterfactual experiments to support our theoretical analysis of the gradient space.

3. Methods

This section introduces the architecture of S2SNets and their signal-preserving training scheme, followed by an empirical evaluation of the gradient space. We test the robustness of S2SNets in a White-Box setting, and compare them to a baseline AE trained with an unsupervised cost function. To evaluate the fulfillment of the second part of Equation 1 without resorting to gradient obfuscation, two experiments are conducted measuring the structural similarity of gradients, and the extent by which adversarial perturbations can be reached by any attacker when relying on gradients from either the classifier or an S2SNet.

3.1. Structure-to-Signal Networks

S2SNets start out with an AE ($\mathbb{A}$) trained with an unsupervised loss $\mathcal{C}_{\mathcal{U}} = \frac{1}{N}\sum_{i=1}^{N} ||x_i - \mathbb{A}(x_i)||_2^2$ on a dataset of $\approx$ 92 million natural images [42]. The architecture of the AE is that of a convolutional autoencoder called SegNet as proposed in [3]. We deliberately select such a large architecture for the AE as we found it to be the closest approximation of the ideal identity function $\mathbb{A}(x) = x$ that does not require shortcut connections between encoder and decoder *e.g.*, as proposed in [32]. This AE is able to reproduce input signals required by both ResNet50 (R) and Inception v3 (I) such that their top-1 accuracy is almost identical the original reported value ($\pm 0.5\%$).

Once the AE has been trained, the fine-tuning of the decoder takes place by freezing the encoder's layers, and then updating only the parameters of the decoder. The update is done via backpropagation of the supervised cross-entropy loss $\mathcal{C}_{\mathcal{X}}$ through the classifier w.r.t. the autoencoded input. In other words, we update the decoder by minimizing the classification loss of the reconstructed samples:

$$\underset{\theta_D}{arg\,min}\, \mathcal{C}_{\mathcal{X}}\big(f(\mathbb{A}(x;\theta_D)), y\big) \tag{2}$$

where θ_D is the set of trainable parameters of the decoder, and f is the pre-trained classifier. Note that the classifier is assumed to already be pre-trained and consequently, its weights remain frozen as well. This training regime has been used in the past to measure the portion of each input sample that is required to achieve state-of-the art performance [27]. We argue that the fine-tuning process imposes an *information bottleneck* that complements the *architectural bottleneck* of the AE, moving the reconstructed samples to areas of the input domain that lay closer to the classification manifold.

Moreover, by fixing the parameters of the encoder after training them on $\mathcal{C}_{\mathcal{U}}$, intermediate representations at the bottleneck of the AE remain class-agnostic. This implies that, during backpropagation through an S2SNet, input gradients explicitly convey low-level structural features only. Semantic information can still be preserved, albeit in a limited way, as long as it relates to said low-level features (*e.g.*, colors of an object, edges between foreground and background). An attacker would still need to differentiate those semantically relevant features from the purely structural ones coming from other areas of the input like the background. Intuitively, we say that input gradients from an S2SNet point to parts of an image that influence the reconstruction error, and not the classification error.

3.2. Threat model

The conditions to test for robustness against adversarial attacks closely follow those from Guo *et al*. [13]:

- **Dataset**: We use ImageNet to test S2SNets in a challenging, large-scale scenario. Classifiers are trained on its training set and evaluations, including attacks, are carried out on the full validation set of 50000 images.
- **Image classifier under attack**: we use ResNet 50 (R) and Inception v3 (I), both pre-trained on ImageNet as target models. They are off-the-shelf models provided by the torchvision project, *i.e.*, they have been trained under clean conditions with no special considerations with respect to adversarial attacks.
- **Defense**: we train an S2SNet for each of the classifiers under attack, following the scheme described in Section 3.1. These defenses are denoted as $\mathbb{A}_R$ and $\mathbb{A}_I$ for S2SAEs fine-tuned on ResNet50 and Inception v3 respectively.
- **Perturbation Magnitude**: we report the normalized L_2 norm [13] between a clean sample x and its adversary $\hat{x} = x + \delta$ denoted as $\mathcal{L}_2(x, \hat{x}) = \frac{1}{N}\sum_{n=1}^{N} \frac{||x_n - \hat{x}_n||_2}{||x_n||_2}$. ϵ-bounds for each attack are listed below.
- **Defense Strength Metric**: to compare classifiers with differing baseline accuracy, vulnerability to adversarial

attacks is measured in terms of the number of newly misclassified samples. More precisely, for any given attack to classifier f, we calculate $\frac{\sum_{x \in TP} \mathcal{L}_2(x, \hat{x})}{|TP|}$, where $TP = \{x \in X | f(x) = y^*\}$ is the set of true positives and $\hat{x}$ is the adversarial example generated by the attack, based on x.

- **Attack Methods**: defended models are tested against a collection of common gradient-based attacks. All samples are cast to the valid integer range $\{0, \dots, 255\}$ for 8-bit RGB images.
  - *Fast Gradient Sign Method (FGSM)* [12]: an effective single-step method that is also fairly transferable to other models. For this method we used $\epsilon \in \{0.5, 1, 2, 4, 8, 16\}$.
  - *Basic Iterative Method (BIM)* [17]: an iterative version of FGSM that shows more adversarial effectiveness but less transferable properties. We used $\epsilon \in \{0.5, 1, 2, 4, 8\}$. The number of iterations is fixed at 10, producing perturbation of up to 10ϵ.
  - *Carlini-Wagner L_2 (CW)* [7]: an optimization-based method that has proven to be effective even against hardened models. Note that this attack typically produces perturbations that lay in a continuous domain. We used $\epsilon \in \{0.5, 1, 2, 4\}$; the number of iterations is fixed at 100, $\kappa = 0$ and $\lambda_f = 10$.
- **Other Attack Conditions**: we test the proposed defenses under a condition known as White-Box. Here, we assume that the attacker has access to the entire ensemble model, *i.e.*, the classifier and the defense. The attacker has therefore access to the predictions of the classifier, intermediate activations, and more importantly, input gradients.

4. Experiments

This section delineates experiments quantifying the robustness of S2SNets and their baselines (Figure 2).

4.1. White-Box

In this experiment, input images flow first through the S2SAE before passing on to the pre-trained classifier. Each classifier requires a separate fine-tuning of an AE (originally trained on $\mathcal{C}_\mathcal{U}$), yielding a different S2SAE per classifier. We use the notation $\mathbb{A}_R$ and $\mathbb{A}_I$ to refer to S2SAEs that have been fine-tuned on ResNet50 and Inception v3 respectively. For an attacker, input gradients come from the first layer of the encoder. Perturbed samples are then passed on again through the whole S2SNet ensemble and the prediction is compared against the original clean sample (Figure 2, bottom). We evaluate against two different baselines: (1) an unprotected network as shown in Figure 2 (top), and (2) an ensemble where the AE has not been fine-tuned on $\mathcal{C}_\mathcal{X}$. To distinguish between fine-tuned AEs and an AE which has only been trained using the unsupervised cost, we add to the latter the subscript "s": $\mathbb{A}_S$. While baseline 1 establishes the vulnerability of a naïve classifier, baseline 2 can be considered to be a large scale evaluation of the defense used in [22]. We contrast the implications of these results with the ones in [22] in subsubsection 4.1.1. Results of White-Box attacks are shown in Figure 3.

We see how undefended models are only able to moderately resist FGSM attacks, while completely failing for the more capable iterative attacks. Surprisingly, the ensemble with $\mathbb{A}_S$ already provides some level of non-trivial robustness against adversarial attacks. For reference, we plot the most comparable result of a defense proposed in [15] and observe that adversarial resiliency is already similar.

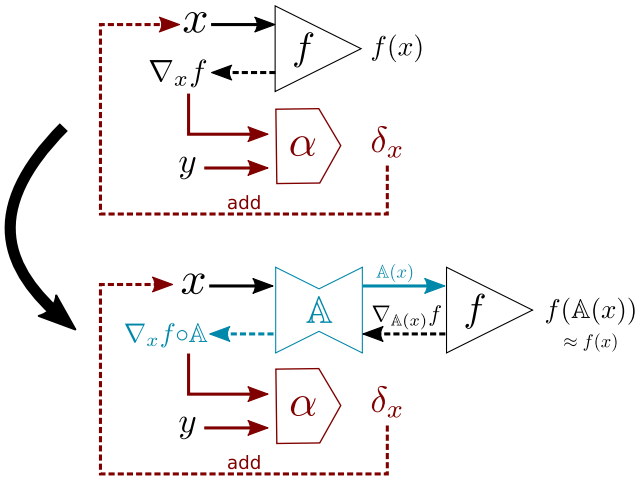

Figure 2: Diagram of the proposed defense mechanism. Top: vulnerable classifier f (black) and attacker α (red). Bottom: same classification scenario with an S2SAE $\mathbb{A}$ defense in place (cyan). To ensure that the defense is indeed effective, $f(x) \approx f(\mathbb{A}(x))$ and $\nabla_x f \cap \nabla_x f \circ \mathbb{A} = \emptyset$ need to hold.

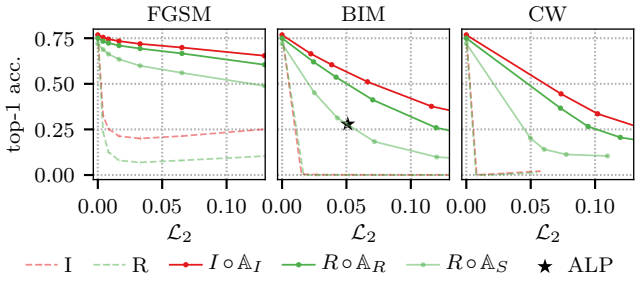

Figure 3: White-Box attacks on ResNet 50 (R) and Inception v3 (I), with (solid) and without (dashed) S2SNet as a defense. Adversarial logit pairing (ALP) [15] added for reference.

When S2SNets are used, we observe a large increase in robustness against all adversarial attacks for all networks. In fact, none of the attacking configurations was able to entirely fool any of the defended classifiers even for the largest ϵ-values that we tested.Note that these white-box results are also similar to ones obtained in [13, 46] albeit under a more forgiving gray-box setting (known classifier but unknown defense).

4.1.1 Adversarial Robustness for $\mathbb{A}_S$

The baseline experiment using $\mathbb{A}_S$ can be compared with the earlier work of Meng *et al*. [22] where it was reported that unsupervised AEs were unsuccessful as a defense against white-box adversarial attacks. However, those experiments were conducted on much smaller datasets: MNIST and CIFAR-10. Our experiments show that, once the problem scales up and both the input space and latent semantic manifold lie in a higher dimensional space, an AE does partially succeed at defending a pre-trained classifier. We argue that this newly discovered resiliency is connected to the presence of much more structurally complex inputs. For ImageNet samples, the distribution of semantically relevant structures (*e.g*., the edges of an airplane) and background structures (*e.g*., the edges of the clouds) will certainly condition which areas of the input produce high gradients when computed w.r.t. $\mathbb{A}_S$. In other words, the *structure-to-signal* ratio increases for tasks like ImageNet while simpler datasets preserve a more balanced ratio[2].

4.2. Properties of the Gradient Space

Experiments from Section 4.1 show that S2SNets add a substantial resiliency to all adversarial attacks in our tests. In this section, we investigate the sources of said adversarial tolerance. In particular, we need to establish that (1) gradients have not been obfuscated *i.e*., they have not simply become unstable, and that (2) the information conveyed by input gradients from an S2SNet are fundamentally different of those from the vulnerable pre-trained classifier.

As defined in [2], testing for gradient obfuscation is done by falsifying the existence of the following conditions:

- *Shattered Gradients*: occurs when the solution is non-differentiable or numerically unstable. S2SNets are end-to-end differentiable through the composition function $f \circ \mathbb{A}$.
- *Stochastic Gradients*: produced by randomized processes in the defense. S2SNets, as proposed in this paper, do not rely on any stochastic process during either the forward, backward, or pre-processing step.
- *Exploding or Vanishing Gradients*: caused by iterative forward passes or networks of increased depth. The composition function $f \circ \mathbb{A}$ adds more layers to the ensemble network. This results in input gradients that are lower in magnitude compared to the classifier on its own. However, FGSM and BIM rely purely on the sign of the gradient and not its norm. A similar re-weighting of the gradient is done for CW. Therefore, since gradients in the 2D input space can be consistently computed, this change in magnitude does not impair the examined attack methods. Experiments supporting this claim are described in the following section.

To further verify that the Structure-to-Signal training scheme produces large shifts in the distribution of input gradients for each sample, we measure the local 2D similarity of the gradients obtained from a vulnerable classifier, the corresponding S2SNet via $\mathcal{C}_{\mathcal{X}}$. We conduct a pairwise evaluation of input gradients coming from the original ResNet50 classifier (R), the original unsupervised AE ($\mathbb{A}_S$), the fine-tuned AE ($\mathbb{A}_R$), an ensemble of classifier and original AE ($R \circ \mathbb{A}_S$), and the proposed S2SNet ensemble ($R \circ \mathbb{A}_R$). For each pair of models we compute the structural similarity [45] (SSIM; a locally normalized mean square error measured in a sliding window) of the input gradients of a single image. Our comparison also includes $\mathbb{A}_s$ and $\mathbb{A}_R$ with gradients for $\mathcal{C}_U$, *i.e*., only the unsupervised reconstruction loss. We then compute the mean of each individual measurement over the whole ImageNet validation set. To stress the differences of the spatial distribution, the SSIM is calculated on the magnitude of the gradients and not on the raw values. Results are summarized in Table 1.

From this experiment we can conclude that input gradients from R are least similar to any other model combination. This supports the claim that the S2SAE changes the spatial distribution of the input gradients. We can also confirm that the distribution change is indeed focused on the low level structure *e.g*., edges, blobs, *etc*. by comparing the higher similarity of $(R \circ \mathbb{A}_R, \mathbb{A}_S)$ compared to $(R \circ \mathbb{A}_R, R)$. Intuitively, this comparison indicates that input gradients

Table 1: Pairwise mean SSIM of input gradient magnitudes for ResNet 50 (R) on the ImageNet validation set, with and without being passed through $\mathbb{A}_R$ or $\mathbb{A}_S$. SSIM values of R w.r.t. any AE variant show the least similarity.

	R	$R \circ \mathbb{A}_R$	$R \circ \mathbb{A}_S$	$\mathbb{A}_R$	$\mathbb{A}_S$
R	1.00	0.17	0.18	0.12	0.14
$R \circ \mathbb{A}_R$	0.17	1.00	0.40	0.46	0.32
$R \circ \mathbb{A}_S$	0.18	0.40	1.00	0.37	0.36
$\mathbb{A}_R$	0.12	0.46	0.37	1.00	0.36
$\mathbb{A}_S$	0.14	0.32	0.36	0.36	1.00

[2]Structure-to-signal ratio, refers to how much semantically relevant information is conveyed by low level features like edges, color, blobs, etc. Non-structural features are, by contrast, higher-level features.

from S2SNets are more closely related to the unsupervised reconstruction objective $\mathcal{C}_{\mathcal{U}}$ than the class-dependent $\mathcal{C}_{\mathcal{X}}$. Evidence of the *representational bottleneck* is also measurable by the large dissimilarity between reconstructions from $\mathbb{A}_S$ and $\mathbb{A}_R$. In fact, the combination of $\mathbb{A}_R$ and $R \circ \mathbb{A}_R$ bear the most similarity, despite the use of $\mathcal{C}_{\mathcal{U}}$ and $\mathcal{C}_{\mathcal{X}}$ respectively to obtain gradients.

As additional baseline, we compare the SSIM of an input gradient w.r.t. its ground-truth label and a randomly selected label. More formally, let x be some input image and y^* its true label. We then randomly select a different label $\hat{y} \neq y^*$ and compute:

$$SSIM(||\nabla_x \mathcal{C}_{\mathcal{X}}(f(x), y^*)||, ||\nabla_x \mathcal{C}_{\mathcal{X}}(f(x), \hat{y})||) \quad (3)$$

for all x in the ImageNet validation set, and $f \in \{R, R \circ \mathbb{A}_R, R \circ \mathbb{A}_S\}$. The SSIM values were: 0.50 for $R \circ \mathbb{A}_R$, 0.47 for $R \circ \mathbb{A}_S$, and 0.34 for R. This shows that the influence of the label is smaller when gradients are propagated through the AE, but also emphasizes how dissimilar gradients of just ResNet50 are, compared to any AE at just 0.12 to 0.18 SSIM (Table 1).

Figure 4 shows the magnitude of the input gradients for the models evaluated in Table 1 to get a more visual intuition of the SSIM results. Here, we observe how gradient magnitudes for AEs ($\mathbb{A}_S$, $\mathbb{A}_R$) predominantly correspond to low level structural features like edges and corners. This is expected, since it is more difficult to accurately reproduce the high frequencies required by sharp edges, compared to the lower frequencies of blobs. Extracting magnitudes based on classification from AEs show similar structures ($R \circ \mathbb{A}_R$, $R \circ \mathbb{A}_S$). Some coincidental overlap between ResNet 50 (R) and AE variants is unavoidable, since edges are also important for classification [48]. However, input gradients from only the classifier are least similar among all model variants. Overall, SSIM is at least twice as high between AE variants, than between the classifier and any of the AE configurations. This evaluation of similarity provides consistent evidence supporting the strong compliance of S2SNets with the second property in Equation 1.

4.2.1 Counterfactual Evaluation of the Structure-to-Signal

One additional experiment to validate our findings comes from using counterfactual thinking. We start by assuming that input gradients of S2SNets do not change when flowing from the classifier to the S2SAE, but rather they are influenced by the forward pass through the S2SAE. In other words, we assume that resiliency to adversarial attacks is not caused by $\nabla_x f \circ \mathbb{A}$ but by the intermediate $\nabla_{\mathbb{A}(x)} f$. If this were the case, crafting attacks based on $\nabla_{\mathbb{A}(x)} f$ and evaluated on the full S2SNet should yield similar results to the ones in Section 4.1. Conversely, perturbations based on $\nabla_x f \circ \mathbb{A}$ should fail when evaluated against the classifier alone.

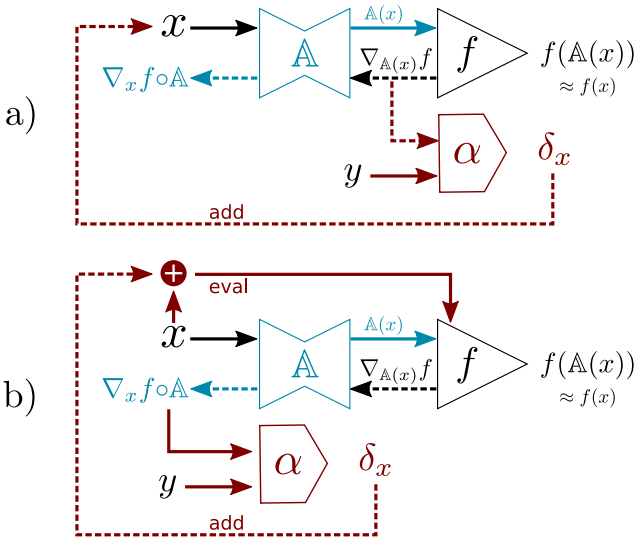

Figure 5: Counterfactual experiments. a) Attacks from gradients at the classifier w.r.t. an input reconstruction, and evaluated on the ensemble S2SNet. b) Attacks based on $\nabla_x f \circ \mathbb{A}$ and evaluated directly on the classifier. If $\mathbb{A}$ is not changing the distribution of input gradients, experiment (a) should fail and experiment (b) should succeed. Neither of these assumptions hold therefore concluding that the S2SAE ($\mathbb{A}$) does change the distribution of gradients.

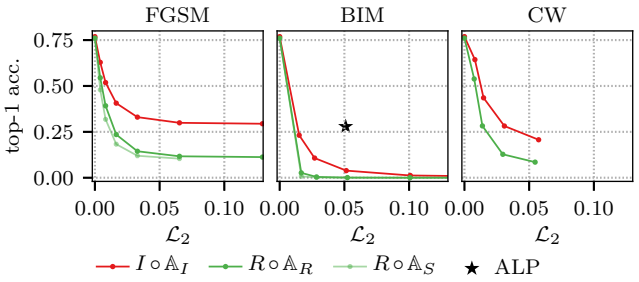

Figure 6: Counterfactual evaluation: perturbations from intermediate input gradients $\nabla_{\mathbb{A}(x)} f$ and evaluated on the ensemble S2SNet. Adversarial logit pairing (ALP) [15] added for reference.

Results for the first counterfactual experiment are shown in Figure 6. Under these conditions, we observe that S2SNet models revert back to the behavior shown by their corresponding unprotected classifiers. In general, this condition is expected, and confirms once more that S2SNets are being trained to preserve the information that is useful to the classifier. Intuitively, gradients collected directly from a vulnerable model convey – by definition – information that relates to the semantic manifold and hence, perturbations based on those gradients will be preserved by the S2SAE.

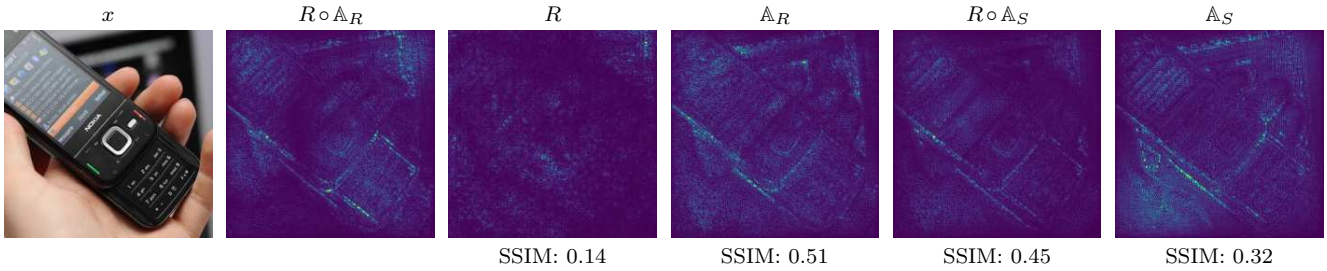

Figure 4: Gradient magnitudes for ResNet 50 (R) given a single input x from the ImageNet validation set, propagated through $\mathbb{A}_R$ or $\mathbb{A}_S$. SSIM values are in comparison to $||\nabla_x R \circ \mathbb{A}_R(x)||$.

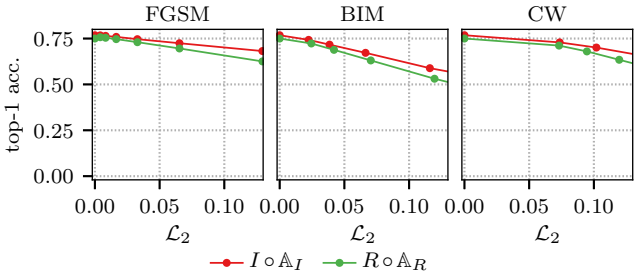

Figure 7: Counterfactual evaluation: perturbations using input gradients from the ensemble S2SNet and evaluated directly on the classifier.

The first counterfactual experiment has therefore failed to produce evidence to disprove the change in the gradient distribution $\nabla_x f \circ \mathbb{A}$.

Results for the second counterfactual experiment Figure 5 (b) are shown in Figure 7 indicate that the attacker is unable to generate perturbations that the classifier alone is vulnerable to. This result in conjunction with the main experiments of section Section 4.1 indicates that input gradients coming from the S2SAE are in a domain where an adversarial attacker cannot reach the range of perturbations that is adversarial for the classifier.

In summary, we show that both counterfactual experiments have failed. Therefore, we conclude that the S2SAE imposes a change in the distribution of the gradients that renders them useless for an adversarial attacker.

5. Conclusions & Future Work

In this paper we presented S2SNets as an effective method to defend pre-trained neural networks against gradient-based adversarial examples. We designed the defense strategy by treating adversarial attacks as functions mapping a domain (gradients from the model) to a range (perturbations for input samples). The defense is based on the notion of both an *architectural bottleneck* and a *representational bottleneck*. While the former is realized through the compression layer of an AE, the latter provides further concentration of the information of each input sample that gets reconstructed. Furthermore, S2SAEs impose a change of distributions in the input gradients they produce due to the use of two cost functions: an unsupervised reconstruction cost for the encoder and a supervised one for the decoder.

We empirically show that S2SNets add non-trivial levels of robustness against white-box adversarial attacks on the full ImageNet validation set. A comparison of the baselines indicates that AEs can already offer resiliency to said attacks, compared to previous claims of the contrary that were based on much smaller datasets.

Experiments to validate the extent of our claims regarding the transformation of input gradients of S2SAEs have been conducted. These include a measurement of structural similarity, visual inspections and two counterfactual hypotheses that were disproved experimentally.

As future work, we would like to explore other ways in which the representational bottleneck can occur. In particular, looking for functions that map back to a different space *i.e.*, $\mathbb{A} : \mathcal{R}^n \rightarrow \mathcal{R}^m$. On the other hand, the preservation of the signal make the S2SAE a promising mechanism to explore and understand the semantic manifold by projecting it back to the input space. We are also interested in analyzing the relationship between the structural complexity of datasets and the ability of AEs to approximate the semantic manifold. Comparing classification consistency of a clean sample, before and after being passed through an S2SNet, has potential implications for detection of adversarial attacks by learning abnormal distribution fluctuations.

Acknowledgments

This work was supported by the BMBF project DeFuseNN (Grant 01IW17002) and the NVIDIA AI Lab (NVAIL) program. We thank all members of the Deep Learning Competence Center at the DFKI for their comments and support.

References

[1] A. Athalye and N. Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

[2] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[3] V. Badrinarayanan, A. Handa, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[4] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[5] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[6] Q.-Z. Cai, M. Du, C. Liu, and D. Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018.

[7] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[8] G. S. Dhillon, K. Azizzadenesheli, J. D. Bernstein, J. Kossaifi, A. Khanna, Z. C. Lipton, and A. Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.

[9] L. Engstrom, A. Ilyas, and A. Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

[10] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. Robustness of classifiers: From adversarial to random noise. In *Advances in Neural Information Processing Systems*, pages 1632–1640, 2016.

[11] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformations. *International Conference on Learning Representations*, 2018. accepted as poster.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[16] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

[17] A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[18] H. Li, Q. Xiao, S. Tian, and J. Tian. Purifying adversarial perturbation with adversarially trained autoencoders. *arXiv preprint arXiv:1905.10729*, 2019.

[19] F. Liao, M. Liang, Y. Dong, T. Pang, J. Zhu, and X. Hu. Defense against adversarial attacks using high-level representation guided denoiser. *arXiv preprint arXiv:1712.02976*, 2017.

[20] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[22] D. Meng and H. Chen. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[23] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *arXiv preprint*, 2017.

[24] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto. Robustness of classifiers to universal perturbations: A geometric perspective. In *International Conference on Learning Representations*, 2018.

[25] S. M. Moosavi Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, pages 3387–3395, 2016.

[27] S. Palacio, J. Folz, J. Hees, F. Raue, D. Borth, and A. Dengel. What do deep networks like to see. In

The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[28] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint*, 2016.

[29] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 582–597. IEEE, 2016.

[30] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie. Generative adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] R. C. Pouya Samangouei, Maya Kabkab. Defensegan: Protecting classifiers against adversarial attacks using generative models. *International Conference on Learning Representations*, 2018. accepted as poster.

[32] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[34] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. Are adversarial examples inevitable? In *International Conference on Learning Representations*, 2019.

[35] A. Sinha, H. Namkoong, and J. Duchi. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

[36] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

[37] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[39] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[40] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[41] T. Tanay and L. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[42] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. The new data and new challenges in multimedia research. *CoRR*, abs/1503.01817, 2015.

[43] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018.

[44] Q. Wang, W. Guo, K. Zhang, I. Ororbia, G. Alexander, X. Xing, C. L. Giles, and X. Liu. Learning adversary-resistant deep neural networks. *arXiv preprint arXiv:1612.01401*, 2016.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[46] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*, 2018.

[47] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[48] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.