Hindering Adversarial Attacks with Implicit Neural Representations

Andrei A. Rusu[1]   **Dan A. Calian**[1]   **Sven Gowal**[1]   **Raia Hadsell**[1]

Abstract

We introduce the Lossy Implicit Network Activation Coding (LINAC) defence, an input transformation which successfully hinders several common adversarial attacks on CIFAR-10 classifiers for perturbations up to $\epsilon = 8/255$ in L_∞ norm and $\epsilon = 0.5$ in L_2 norm. Implicit neural representations are used to approximately encode pixel colour intensities in 2D images such that classifiers trained on transformed data appear to have robustness to small perturbations without adversarial training or large drops in performance. The seed of the random number generator used to initialise and train the implicit neural representation turns out to be necessary information for stronger generic attacks, suggesting its role as a *private key*. We devise a Parametric Bypass Approximation (PBA) attack strategy for key-based defences, which successfully invalidates an existing method in this category. Interestingly, our LINAC defence also hinders some transfer and adaptive attacks, including our novel PBA strategy. Our results emphasise the importance of a broad range of customised attacks despite apparent robustness according to standard evaluations.

1. Introduction

Training Deep Neural Network (DNN) classifiers which are accurate yet generally robust to small adversarial perturbations is an open problem in computer vision and beyond, inspiring much empirical and foundational research into modern DNNs. Szegedy et al. (2014) showed that DNNs are not inherently robust to imperceptible input perturbations, which reliably cross learned decision boundaries, even those of different models trained on similar data. With hindsight, it becomes evident that two related yet distinct design principles have been at the core of proposed defences ever since.

Intuitively, accurate DNN classifiers could be considered robust in practice if: (I) their decision boundaries were largely insensitive to *all* adversarial perturbations, and/or (II) computing *any* successful adversarial perturbations was shown to be expensive, ideally intractable. Early defences built on principle (I) include the adversarial training approach of Madry et al. (2018) and the verifiable defences of Hein & Andriushchenko (2017); Raghunathan et al. (2018), with many recent works continually refining such algorithms, e.g. Cohen et al. (2019); Gowal et al. (2020); Rebuffi et al. (2021). A wide range of defences were built, or shown to operate, largely on principle (II), including adversarial detection methods (Carlini & Wagner, 2017a), input transformations (Guo et al., 2018) and denoising strategies (Liao et al., 2018; Niu et al., 2020). Many such approaches have since been circumvented by more effective attacks, such as those proposed by Carlini & Wagner (2017b), or by using "adaptive attacks" (Athalye et al., 2018; Tramer et al., 2020).

Despite the effectiveness of recent attacks against these defences, Garg et al. (2020) convincingly argue on a theoretical basis that principle (II) is sound; similarly to cryptography, robust learning could rely on computational hardness, even in cases where small adversarial perturbations do exist and would be found by a hypothetical, computationally unbounded adversary. However, constructing such robust classifiers for problems of interest, e.g. image classification, remains an open problem. Recent works have proposed defences based on cryptographic principles, such as the pseudo-random block pixel shuffling approach of AprilPyone & Kiya (2021a). As we will show, employing cryptographic principles in algorithm design is not in itself enough to prevent efficient attacks. Nevertheless, we build on the concept of key-based input transformation and propose a novel defence based on Implicit Neural Representations (INRs). We demonstrate that our Lossy Implicit Neural Activation Coding (LINAC) defence hinders most standard and even adaptive attacks, more so than the related approaches we have tested, without making any claims of robustness about our defended classifier.

Contributions: (1) We demonstrate empirically that lossy INRs can be used in a standard CIFAR-10 image classification pipeline if they are computed using *the same implicit network initialisation*, a novel observation which makes our LINAC defence possible. (2) The seed of the random num-

ber generator used for initialising and computing INRs is shown to be an effective and compact *private key*, since withholding this information hinders a suite of standard adversarial attacks widely used for robustness evaluations. (3) We report our systematic efforts to circumvent the LINAC defence with transfer and a series of adaptive attacks, designed to expose and exploit potential weaknesses of LINAC. (4) To the same end we propose the novel Parametric Bypass Approximation (PBA) attack strategy, valid under our threat model, and applicable to other defences using secret keys. We demonstrate its effectiveness by invalidating an existing key-based defence which was previously assumed robust. (*) Source code will be released before acceptance.

2. Related Work

Adversarial Robustness. Much progress has been made towards robust image classifiers along the adversarial training (Madry et al., 2018) route, which has been extensively explored and is well reviewed, e.g. in (Schott et al., 2019; Pang et al., 2020; Gowal et al., 2020; Rebuffi et al., 2021). While such approaches can be effective against current attacks, a complementary line of work investigates certified defences, which offer guarantees of robustness around examples for some well defined sets (Wong & Kolter, 2018; Raghunathan et al., 2018; Cohen et al., 2019). Indeed, many such works acknowledge the need for complementary approaches, irrespective of the success of adversarial training and the well understood difficulties in combining methods (He et al., 2017). The prolific work on defences against adversarial perturbations has spurred the development of stronger attacks (Carlini & Wagner, 2017b; Brendel et al., 2018; Andriushchenko et al., 2020) and standardisation of evaluation strategies for threat models of interest (Athalye et al., 2018; Croce & Hein, 2020), including adaptive attacks (Tramer et al., 2020). Alongside the empirical progress towards building robust predictors, this line of research has yielded an improved understanding of current deep learning models (Ilyas et al., 2019; Engstrom et al., 2019), the limitations of effective adversarial robustness techniques (Jacobsen et al., 2018), and the data required to train them (Schmidt et al., 2018).

Athalye et al. (2018) show that a number of defences primarily hinder gradient-based adversarial attacks by *obfuscating gradients*. Various forms are identified, such as gradient shattering (Goodfellow et al., 2014), gradient masking (Papernot et al., 2017), exploding and vanishing gradients (Song et al., 2018b), stochastic gradients (Dhillon et al., 2018) and a number of input transformations aimed at countering adversarial examples, including noise filtering approaches using PCA or image quilting (Guo et al., 2018), the Saak transform (Song et al., 2018a), low-pass filtering (Shaham et al., 2018), matrix estimation (Yang et al., 2019)

and JPEG compression (Dziugaite et al., 2016; Das et al., 2017; 2018). Indeed, many such defences have been proposed, as reviewed by Niu et al. (2020), they have ranked highly in competitions (Kurakin et al., 2018), and many have since been shown to be less robust than previously thought, e.g. by Athalye et al. (2018) and Tramer et al. (2020), who use adaptive attacks to demonstrate that several input transformations offer little to no robustness.

To build on such insights, it is worth identifying the "ingredients" essential to the success of adversarial attacks. Most effective attacks, including adaptive ones, assume the ability to approximate the outputs of the targeted model for arbitrary inputs. This is reasonable when applying the correct transformation is tractable for the attacker. Hence, *denying access* to such computations seems to be a promising direction for hindering adversarial attacks. AprilPyone & Kiya (2020; 2021b); MaungMaung & Kiya (2021) borrow standard practice from cryptography and assume that an attacker has full knowledge of the defence's algorithm and parameters, short of a small number of bits which make up a *private key*. Another critical "weakness" of such input denoising defences is that they can be approximated by the identity mapping for the purpose of computing gradients (Athalye et al., 2018). Even complex parametric approaches, which learn stochastic generative models of the input distribution, are susceptible to reparameterisation and Expectation-over-Transformation (EoT) attacks in the white-box setting. Thus, it is worth investigating whether *non-parametric*, *lossy* and *fully deterministic* input transformations exist such that downstream models can still perform tasks of interest to high accuracy, while known and novel attack strategies are either ruled out, or at least substantially hindered, including adaptive attacks.

Implicit Neural Representations. Neural networks have been used to parameterise many kinds of signals, see the work by Sitzmann (2020) for an extensive list, with remarkable recent advances in scene representations (Mildenhall et al., 2020) and image processing (Sitzmann et al., 2020). INRs have been used in isolation per image or scene, not for generalisation across images. Some exceptions exist in unsupervised learning, e.g. Skorokhodov et al. (2021) parameterise GAN decoders such that they directly output INRs of images, rather than colour intensities for all pixels. In this paper we show that INRs can be used to discover functional decompositions of RGB images which enable comparable generalisation to learning on the original signal encoding (i.e. RGB).

3. Hindering Adversarial Attacks with Implicit Neural Representations

In this section we introduce LINAC, our proposed input transformation which hinders adversarial attacks by leverag-
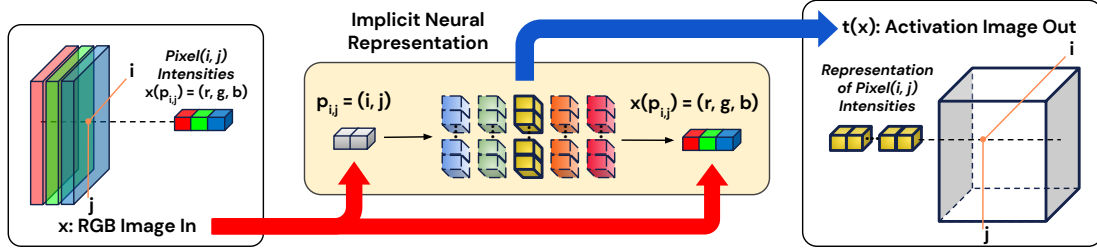
Figure 1. Visual depiction of LINAC, our proposed input transformation. An RGB image x is converted into an *Activation Image* $t(x)$ with identical spatial dimensions, but H channels instead of 3. A neural network model which maps pixel coordinates to RGB colour intensities is fit such that it approximates x. The resulting model parameters (after fitting) are called the Implicit Neural Representation (INR) of image x. In order to output correct RGB colour intensities for all pixels, the implicit neural network needs to compute a hierarchical functional decomposition of x. We empirically choose an intermediate representation to define our transformation. *Activations* in the middle hidden layer are associated with their corresponding pixel coordinates to form the output *Activation Image* $t(x)$, with as many channels as there are units in the middle layer (H).

ing implicit neural representations, also illustrated in Fig. 1.

Setup. We consider a supervised learning task with a dataset $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ of pairs of images x and their corresponding labels y. We use a deterministic input transformation $t \colon \mathcal{X} \to \mathcal{H}$ which transforms input images, $x \mapsto t(x)$, while preserving their spatial dimensions. Further, we consider a classifier f_θ, parameterised by θ, whose parameters are estimated by Empirical Risk Minimisation (ERM) to map transformed inputs to labels $f_\theta \colon \mathcal{H} \to \mathcal{Y}$. The model is not adversarially trained, yet finding adversarial examples for it is hindered by LINAC, as we demonstrate through extensive evaluations in Section 5.

Implicit Neural Representations. For an image x, its implicit neural representation is given by a multi-layer perceptron (MLP) $\Phi = h^L \circ h^{L-1} \circ \cdots \circ h^0$, $\Phi \colon \mathbb{R}^2 \to \mathbb{R}^3$, with L hidden layers, which maps spatial coordinates to their corresponding colours. Φ_ϕ is a solution to the implicit equation:

$$\Phi(\mathbf{p}) - x(\mathbf{p}) = 0, \qquad (1)$$

where $\mathbf{p}$ are spatial coordinates (i.e. pixel locations) and $x(\mathbf{p})$ are the corresponding image colours. Our input transformation leverages this implicit neural representation to encode images in an approximate manner.

Reconstruction Loss. The implicit equation (1) can be translated (Sitzmann et al., 2020) into a standard reconstruction loss between image colours and the output of a multi-layer perceptron Φ_ϕ at each (2D) pixel location $\mathbf{p}_{i,j}$,

$$\mathcal{L}(\phi, x) = \sum_{i,j} ||\Phi_\phi(\mathbf{p}_{i,j}) - x(\mathbf{p}_{i,j})||_2^2. \qquad (2)$$

We provide pseudocode for the LINAC transform in Algorithm 1 and a discussion of computational and memory requirements in Appendix A.1.4. For each individual image x, we estimate $\hat{\phi}_x$, an approximate local minimiser of

---

Algorithm 1 The LINAC Transform

Inputs: RGB image x (with size $I \times J \times 3$); *private key*; number of epochs N; mini-batch size M; number of MLP layers L; representation layer K; learning rate μ.
Output: Activation Image $t(x)$ (with size $I \times J \times H$).
$\text{rng} = \text{INIT\_PRNG}(\textit{private key})$ ▷ Seed rng.
$\phi^{(0)} = \text{INIT\_MLP}(\text{rng}, L)$
$S = \lfloor I \cdot J / M \rfloor$ ▷ Num. mini-batches per epoch.
$\phi = \phi^{(0)}$
for $epoch = 0 \dots N - 1$ **do**
 $\mathcal{P} = \text{SHUFFLE\_AND\_SPLIT\_PIXELS}(x, \text{rng}, S)$
 for $m = 0 \dots S - 1$ **do**
  $\ell = \frac{1}{M \cdot I \cdot J} \sum_{(i,j) \in \mathcal{P}[m]} ||\Phi_\phi(\mathbf{p}_{i,j}) - x(\mathbf{p}_{i,j})||_2^2$
  $\phi = \phi - \mu \nabla_\phi \ell$
 end for
end for
$\hat{\phi}_x = \phi$
Return $t(x)$ applying Eq. 3 using $\hat{\phi}_x$ and layer K.

$\mathcal{L}(\phi, x)$, using a stochastic iterative minimisation procedure with mini-batches of M pixels grouped into epochs, which cover the entire image in random order, for a total of N passes through all pixels.

Private Key. A random number generator is used for: (1) generating the initial MLP parameters $\phi^{(0)}$ and (2) for deciding which random subsets of pixels make up mini-batches in each epoch. This random number generator is seeded by a 64-bit integer which we keep secret and denote as the *private key*. Hence, for all inputs x we start each independent optimisation from the same set of initial parameters $\phi^{(0)}$, and we use the same shuffling of pixels across epochs.

Lossy Implicit Network Activation Coding (LINAC). We consider the lossy encoding of each pixel (i, j) in image x as the H-dimensional intermediate activations vector of layer K of the MLP evaluated at that pixel position:

$c_x(i,j) = (h_{\hat{\phi}_x}^{K-1} \circ \cdots \circ h_{\hat{\phi}_x}^0)(\mathbf{p}_{i,j})$ with $K < L$. We build the *lossy implicit network activation coding* transformation of an image x by stacking together the encodings of all its pixels in its 2D image grid, concatenating on the feature dimension axis. The LINAC transformation $t(x)$ of the $I \times J \times 3$ image x is given by:

$$t(x) = \begin{bmatrix} c_x(0,0) & \dots & c_x(0, J-1) \\ \vdots & \ddots & \vdots \\ c_x(I-1, 0) & \dots & c_x(I-1, J-1) \end{bmatrix}, \quad (3)$$

and has dimensionality $I \times J \times H$, where H is the number of outputs of the K-th layer of the MLP. By construction, our input transformation preserves the spatial dimensions of each image while increasing the feature dimensionality (from 3, the image's original number of colour channels, to H); this means that standard network architectures used for image classification (e.g. convolutional models) can be readily trained as the classifier f_θ.

All omitted implementation details are provided in Appendix A, and sensitivity analyses of LINAC to its hyperparameters are reported in Appendix C.

Threat Model. We are interested in hindering adversarial attacks on a nominally-trained classifier $f_\theta(t(x))$, which operates on transformed inputs (i.e. on $t(x)$ rather than on x), using a *private key* of our choosing. Next, we describe the threat model of interest by stating the conditions under which the LINAC defence is meant to hinder adversarial attacks on f_θ, following AprilPyone & Kiya (2021a).

We assume attackers do not have access to the *private key*, the integer seed of the random number generator used for computing the LINAC transformation, but otherwise have full algorithmic knowledge about our approach. Specifically, we assume an attacker has complete information about the classification pipeline, including the architecture, training dataset and weights of the defended classifier. This includes full knowledge of the LINAC algorithm, the implicit network architecture, parameter initialisation scheme and all the fitting details, except for the *private key*.

4. Attacking the LINAC Defence

Setup. We are interested in evaluating the apparent robustness of a LINAC-defended classifier, $f_{\hat{\theta}}$, which has been trained by ERM to classify transformed inputs from the dataset $\mathcal{D}$. Specifically, its parameters $\hat{\theta}$ minimise $\mathbb{E}_{x,y \sim \mathcal{D}}[\mathcal{L}_{\text{CE}}(f_\theta(t(x)), y)]$, where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and $t(x)$ is the LINAC transformation applied to image x using the *private key*.

Input Perturbations. Classifiers defended by LINAC are not adversarially trained (Madry et al., 2018) to increase their robustness to specific L_p norm-bounded input pertur-

bations. Furthermore, the LINAC defence is inherently agnostic about particular notions of maximum input perturbations. Nevertheless, to provide results comparable with a broad set of defences from the literature, we perform evaluations on standard L_p norm-bounded input perturbations with: (1) a maximum perturbation radius of $\epsilon = 8/255$ in the L_∞ norm, and (2) one of $\epsilon = 0.5$ in the L_2 norm.

Adapting Existing Attacks. Without access to the *private key* an attacker cannot compute the LINAC transformation exactly. However, an attacker could acquire access to model inferences by attempting to brute-force guess the *private key*. Another option would be to train surrogate models with LINAC, but using keys chosen by the attacker, in the hope that decision boundaries of these models would be similar enough to mount effective transfer attacks. More advanced attackers could modify LINAC itself to enable strong Backward Pass Differentiable Approximation (BPDA) (Athalye et al., 2018) attacks. We evaluate the success of these and other standard attacks in Section 5.

Designing Adaptive Attacks. Athalye et al. (2018) provide an excellent set of guidelines for designing and performing successful adaptive attacks, while also standardising results reporting and aggregation. Of particular interest for defences based on input transformations are the BPDA and Expectation-over-Transformation (EoT) attack strategies. Subsequent work convincingly argues that adaptive attacks are not meant to be general, and must be customised, or "adapted", to each defence in turn (Tramer et al., 2020). While BPDA and EoT generate strong attacks on input transformations, they both rely on being able to compute the forward transformation or approximate it with samples. Indeed, the authors mention that substitution of both the forward and backward passes with approximations leads to either completely ineffective, or much less effective attacks.

Parametric Bypass Approximation (PBA). Inspired by the reparameterisation strategies of Athalye et al. (2018), we propose a bespoke attack by making use of several pieces of information available under our threat model: the parametric form of the defended classifier $f_\theta(t(x))$, its training dataset $\mathcal{D}$ and loss function $\mathcal{L}_{\text{CE}}$, and its trained weights $\hat{\theta}$.

A Parametric Bypass Approximation of an unknown nuisance transformation $u : \mathcal{X} \to \mathcal{H}$ is a surrogate parametric function $h_\psi : \mathcal{X} \to \mathcal{H}$, parameterised by a solution to the following optimisation problem:

$$\psi^* = \arg\min_\psi \mathbb{E}_{x,y \sim \mathcal{D}} \left[\mathcal{L}_{\text{CE}}(f_{\hat{\theta}}(h_\psi(x)), y) \right]. \quad (4)$$

This formulation seeks a set of parameters ψ^* which minimise the original classification loss while keeping the defended classifier's parameters frozen at $\hat{\theta}$. Similar with classifier training, this optimisation problem can be solved efficiently using Stochastic Gradient Descent (SGD).

A PBA adversarial attack can then proceed by approximating the *defended classifier* $f_{\hat{\theta}}(u(\cdot))$ with those of the *bypass classifier* $f_{\hat{\theta}}(h_{\psi^*}(\cdot))$ in both forward and backward passes when computing adversarial examples, e.g. using Projected Gradient Descent (PGD).

The main advantages of the PBA strategy are that no forward passes through the nuisance transformation $u(\cdot)$ are required, and that it admits efficient computation of many attacks to $f_{\hat{\theta}}$, including gradient-based ones. In Section 5 we demonstrate the effectiveness of PBA beyond the LINAC defence. We show that, even though the surrogate transformation is fit on training data only, the defended classifier operating on samples passed through h_{ψ^*} (bypassing u) demonstrates nearly identical generalisation to the test set. Furthermore, we also show that PBA has greater success at finding adversarial examples for the LINAC defence compared to other methods. Lastly, we use PBA to invalidate an existing key-based defence proposed in the literature.

5. Results

5.1. Evaluation Methodology

Since LINAC makes no assumptions about adversarial perturbations, we are able to evaluate *a single defended classifier model* against all attack strategies considered, in contrast to much adversarial training research (Madry et al., 2018).

To obtain a more comprehensive picture of apparent robustness we start from the rigorous evaluation methodology used by Gowal et al. (2019); Rebuffi et al. (2021). We perform untargeted PGD attacks with 100 steps and 10 randomised restarts, as well as multi-targeted (MT) PGD attacks using 200 steps and 20 restarts. Anticipating the danger of obfuscated gradients skewing results, we also evaluate with the Square approach of Andriushchenko et al. (2020), a powerful gradient-free attack, with 10000 evaluations and 10 restarts. For precise comparisons with the broader literature we also report evaluations using the parameter-free AutoAttack (AA) strategy of Croce & Hein (2020).

Following Athalye et al. (2018) we aggregate results across attacks by only counting as accurate robust predictions those test images for which the defended classifier predicts the correct class with and without adversarial perturbations, computed using all methods above. We report this as *Best Known* robust accuracy.

In instances where several surrogate models are used to compute adversarial perturbations, also known as *transfer attacks*, we report *Best Adversary* results aggregated for each individual attack, which is defined as robust accuracy against all source models considered.

We aggregate evaluations across these two dimensions (attacks & surrogate models) by providing a single robust ac-
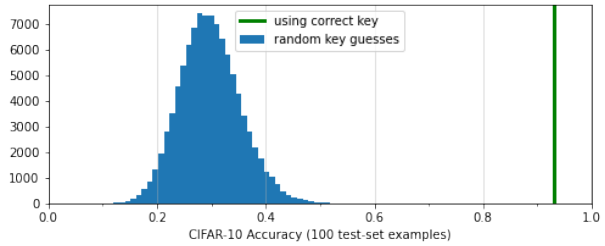

Figure 2. Results of direct attack on *private key*. A histogram of accuracies of the same defended classifier with inputs transformed using either the correct key or 100000 randomly chosen keys. An appropriate surrogate transformation is not found, invalidating attack vectors which rely on access to the outputs of the defended model on attacker chosen inputs.

curacy number against all attacks computed using all source models for each standard convention of maximum perturbation and norm, enabling easy comparisons with results in the literature.

5.2. Attacks with Surrogate Transformations & Models

A majority of adversarial attack strategies critically depend on approximating the outputs of the defended classifier for inputs chosen by the attacker. The *private key* is kept secret in our threat model, which means that an attacker can neither compute the precise input transformation used to train the defended classifier, nor its outputs on novel data. Hence, an attacker must find appropriate surrogate transformations, or surrogate classifier models, in order to perform effective adversarial attacks. We investigate both strategies below.

Firstly, we empirically check that the outputs of the defended classifier cannot be usefully approximated without knowledge of the *private key*. It is reasonable to hypothesise that transformations with different keys may lead to similar functional representations of the input signal. We start investigating this hypothesis by simply computing the accuracy of the defended model on clean input data transformed with LINAC, but using keys chosen by the attacker, also known as a brute-force key attack, which is valid under our threat model. As reported in Figure 2, the accuracy of our LINAC defended classifier on test inputs transformed with the correct *private key* is over 93%. In an attempt to find a surrogate transformation, 100000 keys are picked uniformly at random. For each key, we independently evaluated the accuracy of the classifier using a batch of 100 test examples, and we report the resulting accuracy estimates for all keys with a histogram plot. The mean accuracy with random key guesses is around 30%, with a top accuracy of just 57% (see Table 4 in Appendix B.1 for a breakdown). Hence, using LINAC with incorrect keys leads to poor approximations of classifier outputs on correctly transformed data. This suggest that the learned decision boundaries of the defended classifier are not invariant to the *private key* used by LINAC.

While we could not find a useful surrogate transformation

			Transfer Attack Source Models				Best Adversary
Norm	Attack	Nominal Source	Adversarial Training (L_∞)	Adversarial Training (L_2)	Defended Surrogates (Attacker Keys)	Reconstruction-Based Surrogates (BPDA)	All Source Models
L_∞	AA	92.77	80.42	70.29	84.00	59.40	41.18
	MT	84.57	72.96	56.08	85.70	55.37	47.91
	PGD	85.99	60.97	44.06	87.32	56.00	41.22
	Square	85.12	65.69	52.66	75.91	69.14	49.76
	Best Known	81.91	54.97	39.20	75.64	51.17	37.04
L_2	AA	90.84	86.75	80.83	88.27	74.59	71.32
	MT	87.55	85.34	84.81	87.31	74.98	73.83
	PGD	88.61	82.39	74.19	88.36	75.00	70.90
	Square	88.58	84.50	79.31	84.08	83.26	77.68
	Best Known	86.06	79.42	71.92	83.48	71.89	68.41

Table 1. CIFAR-10 test set robust accuracy (%) of a single LINAC defended classifier according to a suite of L_∞ and L_2 transfer attacks, valid under our threat model, using various source classifiers to generate adversarial perturbations.
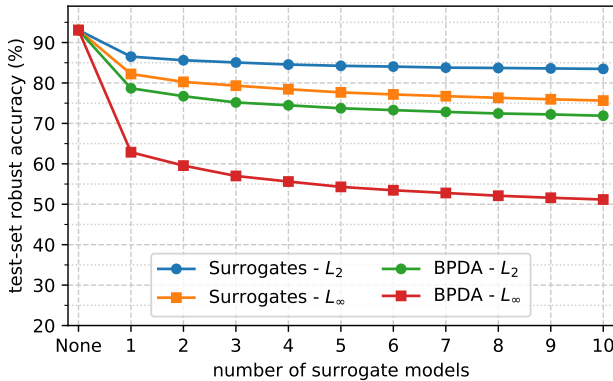

Figure 3. Plots of CIFAR-10 test-set robust accuracy estimates (*Best Known*) vs. number of attacker-trained surrogate models. We also plot the clean accuracy of 93.08% for reference.

by random guessing, it is still possible that transformations with different keys preserve largely the same input information. So, the second option of an attacker is to check whether decision boundaries of models defended with LINAC and different keys are in fact very similar, which would enable powerful transfer attacks from such *surrogate models*. To this end, 10 independent models defended with LINAC were trained from scratch, each using a different key chosen by the attacker. We used the most promising 10 keys from the brute-force key attack for this purpose. In Figure 3 we report *Best Known* robust accuracy plotted against the number of surrogate models used in these joint attacks, and we aggregate results over all 10 attacking models in the fourth column of Table 1. However, this attack vector has limited success. Under transfer attacks with such surrogate models, the robust accuracy of our defended classifier appears to be high. While PGD and MT may fail due to vanishing or exploding gradients (Athalye et al., 2018), Square is a gradient-free attack, and does not suffer from such issues. Robust accuracy estimates according to Square are higher than 83% against any individual surrogate model, irrespective of perturbation norm. A complete breakdown of results is given in Table 5 of Appendix B.1. Attacking with all 10

surrogate models together, robust accuracy to Square is still higher that 75%, and the estimate is not improved by further aggregating over attacks. This evidence further support the hypothesis that decision boundaries of classifiers defended with LINAC depend on their respective keys, and may differ enough across keys to hinder transfer attacks with surrogates. Investing an order of magnitude more computation into such attacks leads to modest reductions in apparent robustness.

Lastly, an attacker may strive to employ BPDA, one of the most effective and general strategies against defences using nuisance transformations. BPDA attacks require: (1) the ability to compute the exact forward transformation and (2) finding a usefully differentiable approximation to the said transformation for use in the backwards pass of gradient-based attacks. In many cases this would be enough to allow the attacker to compute adversarial examples, perhaps at a somewhat higher computational cost (Athalye et al., 2018; Tramer et al., 2020).

Our LINAC defence presents further challenges by design. Exact forward computations (model inferences) require the *private key*. An attacker cannot exactly compute the input transformation even for training set images, e.g. in order for some differentiable parametric approximation to be learned in a supervised fashion. Furthermore, surrogate models defended using LINAC and attacker chosen keys do not appear to be usefully differentiable, as suggested by results in Table 1. Nevertheless, an attacker could still hope that our defence "filters out" information in a largely key-agnostic manner, and that the choice of implicit network representation layer is not essential. Hence, they have the option of modifying LINAC to output activations of the last, rather than the middle layer of the implicit network. This amounts to reducing LINAC to an approximate reconstruction of the original signal. While such surrogate models with attacker chosen keys would still have to be trained for the purpose, they would be vulnerable to strong BPDA attacks,

| | | **All Source Models** | **Adaptive Attacks** | |
Norm	Attack	Transfer	BPDA	PBA
L_∞	AA	41.18	59.40	68.34
	MT	47.91	55.37	46.75
	PGD	41.22	56.00	44.05
	Square	49.76	69.14	48.59
	Best Known	37.04	51.17	35.32
L_2	AA	71.32	74.59	73.10
	MT	73.83	74.98	67.85
	PGD	70.90	75.00	66.93
	Square	77.68	83.26	74.70
	Best Known	68.41	71.89	61.23

Table 2. CIFAR-10 test set robust accuracy (%) of a single LINAC defended classifier w.r.t. a suite of L_∞ and L_2 attacks, valid under our threat model, using different strategies such as transfer and adaptive attacks. Our novel PBA adaptive attacks are overall more effective that both transfer and BPDA attack strategies.

which may transfer well to our defended classifier. Apparent robustness estimates according to such transfer BPDA attacks are plotted in Figure 3 as a function of the number of surrogate models used jointly in the attack. In the fifth column of Table 1 we provide aggregate apparent robust accuracies using 10 such surrogates, showing that transfer BPDA attacks are more successful than previous attempts; any such reconstruction-based surrogate model can be used to reveal that the robust accuracy of our defended classifier cannot be higher than 65%, particularly with standard L_∞ multi-targeted (MT) attacks (see Table 6 in Appendix B.1 for a detailed breakdown of results). Interestingly, when 10 surrogate models are used together, L_∞ robust accuracy estimates drop to 51%. The reduction is less severe in standard L_2 attacks, where accuracy against all surrogates appears to be still over 71%. These results confirm that the BPDA strategy is a valuable tool for investigating the robustness of a wide range of defences, even when its assumptions are not fully met.

5.3. Transfer Attacks with Nominal and Adversarially Trained Source Models

Since our defended classifier is not adversarially trained, one could assume that its decision boundaries may be similar to those of a nominal, undefended classifier. We show in the first column of Table 1 that transfer attacks with a nominally trained source model have limited success, especially considering that such undefended classifiers have below chance robust accuracies according to the very same evaluations.

Another possibility is that that our defended model may be susceptible to the promising attack directions to which adversarially trained robust classifiers are vulnerable. We report in the second and third columns of Table 1 that this is indeed the case to some extent. Of all adversaries considered thus far, a robust model adversarially trained to tolerate perturbations of up to size $\epsilon = 0.5$ in L_2 norm leads to the

most effective transfer attacks. This holds to a lesser degree for an adversarially trained model with perturbations of size $\epsilon = 8/255$ in L_∞ norm. Despite the success of evaluations using the former source model, no one attack method comes close to the effectiveness of the joint strategy, reported as *Best Known* robust accuracy.

Furthermore, it is important to note that ensemble transfer attacks are much stronger than those computed with any given source model. Aggregated over four attack types and 23 different source models, the robust accuracy of our LINAC defended classifier is revealed to be at most half of what initial results suggested according to aggregate L_∞ evaluations; this does not appear to be the case for L_2 attacks, however, which continue to be substantially hindered by LINAC. Robust accuracy could still be above 68% according to the latter attack type, even in aggregate. In order to better characterise the implications of LINAC we make use of novel adaptive attacks in the following subsection.

5.4. PBA Attacks Against LINAC

Thus far we have shown that strong transfer attacks can be performed by using an ensemble of diverse source models to compute adversarial perturbations over many repeated trials. While ultimately more reliable, this is a cumbersome evaluation protocol, requiring two order of magnitude more computation than standard evaluations.

In Section 4 we have introduced PBA, an attack strategy purposefully designed to be effective against input transformations (or network modules) which deny both inference and gradient computations, despite classifier parameters, training loss and dataset being available to the attacker. Following this novel strategy we successfully trained a parametric bypass approximation (PBA) of the LINAC transform and its associated *defended classifier*. Intriguingly, the decision boundaries of the resulting *bypass classifier* generalise very well. Accuracy on clean test data is 95.35%. Furthermore, the *bypass classifier* can be readily shown to have 0% robust accuracy using PGD attacks. This indicates that any apparent robustness in evaluations can be largely attributed to the LINAC transform successfully hindering attacks, since the decision boundaries of our *defended classifier* are susceptible to adversarial perturbations, and hence cannot be considered to add any inherent robustness by themselves.

In Table 2 we show that standard attacks using the trained PBA mapping against our LINAC defended classifier are even more effective than BPDA attacks using 10 source models. Interestingly, PBA almost uniformly leads to more effective attacks, regardless of strategy. PGD attacks using PBA give the most accurate picture of robustness of all strategies, suggesting that the matter of *obfuscated gradients* is largely mitigated by our novel strategy. Aggregated over different attack types, PBA is the most effective and
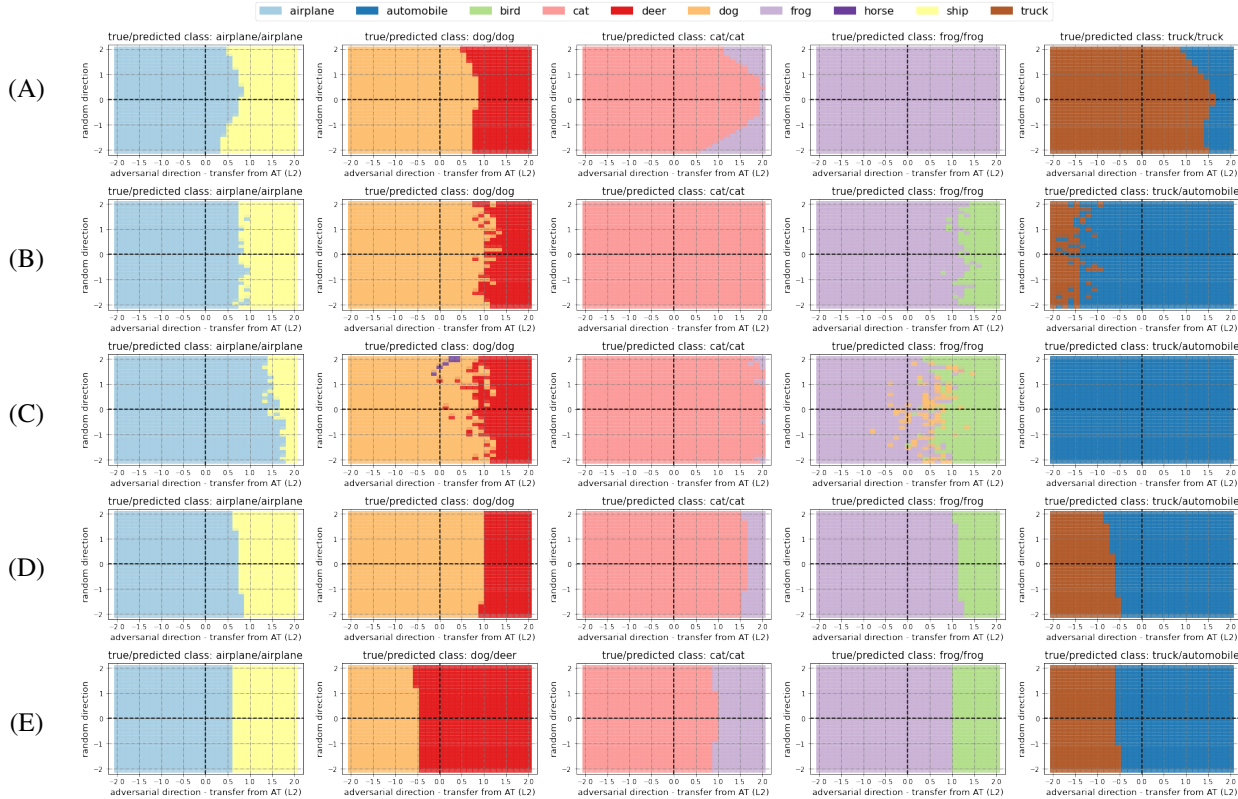
Figure 4. Decision boundaries of five different classifiers (rows) around the same five randomly chosen test examples (columns), plotted along their respective adversarial directions according to the AT (L_2) model (horizontal), and the same random direction (vertical): (A) An undefended, nominally trained CIFAR-10 classifier; (B) LINAC defended classifiers using a random key; (C) LINAC defended classifiers using the *private key*; this is the model we evaluated throughout the submission; (D) The by-pass classifier resulting from our novel PBA attack on model (C); (E) An adversarially trained classifier, AT (L_2) in the main text, used to generate transfer attacks. We observe that boundaries of nominal model (A) are different from those of LINAC (B) & (C); LINAC decision boundaries seem less smooth compared to other models; as suspected boundaries appear different across keys (B) vs. (C), which corroborates our observations of robustness to transfer attacks with surrogates. The adversarially trained model (E) is robust to the vertical dimension (random noise); LINAC models (B) & (C) also appear less sensitive to random noise compared the nominal model (A). PBA by-pass classifier (D) boundaries are much smoother and different from the true boundaries of the attacked model (C), which may explain why LINAC withstands the novel attack in many cases. Notice that PBA approximated boundaries (D) can be both closer and farther away from test examples compared to the true model's (C), which makes it less clear how useful such approximations are for future attacks on LINAC.

efficient evaluation strategy which does not make use of the *private key*, and hence is valid under the adopted threat model. Based on these evaluations alone, one may conclude that robust accuracy was over 35% under attacks of size at most $\epsilon = 8/255$ in L_∞ norm, and over 61% for attacks of size $\epsilon = 0.5$ in L_2 norm. The apparent robustness difference between L_∞ and L_2 attacks persists, suggesting that LINAC primarily hinders the latter type of attacks.

5.5. Towards Explaining the Apparent Robustness

Decision boundary inspection. We plotted decision boundaries of several classifiers around five randomly chosen test examples in Figure 4. All boundary plots are centred on test examples (columns), use appropriate adversarial directions as the horizontal dimension, and a random direction as

the vertical. As expected, we observe differences between LINAC defended classifiers which use different keys. Furthermore, we found that LINAC boundaries can be more "complicated" relative to those of other models, which may explain why PBA attacks are not completely effective.

RGB Reconstruction vs. Lossy Encodings. Setting the representation layer index $K = L$ renders our LINAC transform into an approximate RGB input reconstruction, since L is the index of the implicit network output layer. We confirmed that setting $K = L = 5$ and $N = 100$ epochs offers no robustness, since the resulting reconstructions are precise and BPDA attacks are successful. Clean accuracy was 96.91%, virtually matching that of a nominally trained classifier. Hence, any apparent robustness must be due to the number of INR fitting epochs N, and/or the choice of repre-

	Full PCA	Block PCA	JPEG (23)	JPEG (10)	Block Pixel Shuffle	LINAC (Ours)
Clean Accuracy:	96.10	96.39	88.15	81.17	96.98	93.08

Norm	Attack	Standard	Standard	BPDA	BPDA	PBA	PBA
L_∞	AA	0.00	0.00	11.90	32.58	0.18	68.34
	MT	0.00	0.00	–	–	0.00	46.75
	PGD	0.00	0.00	17.49	27.48	0.00	44.05
	Square	0.00	0.00	5.36	6.34	0.00	48.59
	Best Known	0.00	0.00	0.61	2.26	0.00	35.32
L_2	AA	0.00	0.06	62.95	62.98	0.02	73.10
	MT	0.03	0.00	–	–	0.00	67.85
	PGD	0.41	0.17	60.37	60.38	0.02	66.93
	Square	12.85	11.66	28.33	21.92	6.13	74.70
	Best Known	0.02	0.00	14.94	14.56	0.00	61.23

Table 3. CIFAR-10 test set robust accuracy (%) of several classifiers defended using related input transformations according to evaluations using adversarial perturbations bounded in L_∞ and L_2 norms. Reporting results of strongest known attack strategy for each method, valid according to its own threat model.

sentation layer index K. Intuitively, both hyper-parameters control how "lossy" our transformation is.

Naturally, we were interested in reducing the computational overhead of LINAC. Aiming to match the clean accuracy of state-of-the-art adversarially trained robust classifiers, specifically 93% (Rebuffi et al., 2021), we empirically chose $N = 10$ epochs as a trade-off between speed and clean accuracy. The activation coding layer index $K = 2$ out of $L = 5$ hidden layers was chosen according to the same principle, as the lowest level representation which did not reduce clean accuracy below the target threshold. We further characterise and illustrate our LINAC transform in Appendix D.

Performance Considerations. LINAC is as expensive as inference with a WideResNet-70-16 (Zagoruyko & Komodakis, 2016) on CIFAR-10 images. This cost is dominated by the fitting of INRs. It could be reduced with an adaptive form of "early stopping" based on loss values, or by leveraging advances in INR research (e.g. Sitzmann et al. (2020)). We leave these investigations, and scaling LINAC to larger images, for future work.

Sensitivity Analyses. The apparent robustness of LINAC defended classifiers is largely insensitive to the number of hidden layers $L \geq 3$ of the implicit MLP, as well as the number of features $F \geq 3$ in its positional input encoding, hence we relegated the sensitivity analyses to Appendix C.

5.6. PBA Beyond LINAC and Methodology Validation

We show in the one-but-last column of Table 3 that PBA successfully and completely invalidates the Block Pixel Shuffle approach of AprilPyone & Kiya (2021a), despite its good reported robustness against all attacks. We further investigate using adversarially trained source models, see

full results in Table 7 of Appendix B.1. In summary, our analysis confirms that the apparent robust accuracy of Block Pixel Shuffle according to valid attacks bounded in L_2 norm remains high at 69%. Hence, PBA is indeed the only known valid attack on this defence which is completely successful.

Finally, we validate our evaluation methodology by testing its effectiveness against similar defences. We perform the same evaluations on the Principal Component Analysis (PCA) based defence of Shaham et al. (2018), and the JPEG-based defences of Das et al. (2017; 2018); Guo et al. (2018). In Table 3 we report the *Best Known* robust accuracies of these defences according to our evaluation methodology, which are directly comparable with our reported LINAC results. We observe that LINAC successfully hinders much stronger attacks than these alternative strategies.

6. Conclusions

In this work we introduce LINAC, a novel key-based defence using implicit neural representations, and demonstrate its effectiveness for hindering standard adversarial attacks on CIFAR-10 classifiers. We systematically attempt to circumvent our defence by adapting a host of widely used attacks from the literature, including transfer and adaptive attacks, but LINAC maintains strong apparent robustness. Consequently, we challenge LINAC by introducing a novel adaptive attack strategy (PBA) which is indeed more successful at discovering adversarial examples. We also show that PBA can be used to completely invalidate an existing key-based defence. These are some of the latest attempts to leverage computational hardness for adversarial robustness, and successful PBA attacks on existing methods enable further progress.

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

AprilPyone, M. and Kiya, H. An extension of encryption-inspired adversarial defense with secret keys against adversarial examples. In *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1369–1374. IEEE, 2020.

AprilPyone, M. and Kiya, H. Block-wise image transformation with secret key for adversarially robust defense. *IEEE Transactions on Information Forensics and Security*, 16:2709–2723, 2021a.

AprilPyone, M. and Kiya, H. Transfer learning-based model protection with secret key. *arXiv preprint arXiv:2103.03525*, 2021b.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs. 2018. URL http://github.com/google/jax.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 3–14, 2017a.

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

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

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

Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Chen, L., Kounavis, M. E., and Chau, D. H. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *arXiv preprint arXiv:1705.02900*, 2017.

Das, N., Shanbhogue, M., Chen, S.-T., Hohman, F., Li, S., Chen, L., Kounavis, M. E., and Chau, D. H. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 196–204, 2018.

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

Dziugaite, G. K., Ghahramani, Z., and Roy, D. M. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., and Madry, A. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Frostig, R., Johnson, M. J., and Leary, C. Compiling machine learning programs via high-level tracing. *Systems for Machine Learning*, 2018.

Garg, S., Jha, S., Mahloujifar, S., and Mohammad, M. Adversarially robust learning could leverage computational hardness. In *Algorithmic Learning Theory*, pp. 364–385. PMLR, 2020.

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

Gowal, S., Uesato, J., Qin, C., Huang, P., Mann, T. A., and Kohli, P. An alternative surrogate loss for pgd-based adversarial testing. *CoRR*, abs/1910.09338, 2019. URL http://arxiv.org/abs/1910.09338.

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

Guo, C., Rana, M., Cisse, M., and van der Maaten, L. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

He, W., Wei, J., Chen, X., Carlini, N., and Song, D. Adversarial example defenses: ensembles of weak defenses are not strong. In *Proceedings of the 11th USENIX Conference on Offensive Technologies*, pp. 15–15, 2017.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2263–2273, 2017.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Jacobsen, J.-H., Behrmann, J., Zemel, R., and Bethge, M. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pp. 195–231. Springer, 2018.

Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., and Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser, 2018.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. URL http://arxiv.org/abs/1608.03983.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

MaungMaung, A. and Kiya, H. A protection method of trained cnn model with secret key from unauthorized access. *arXiv preprint arXiv:2105.14756*, 2021.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision (ECCV)*, pp. 405–421. Springer, 2020.

Niu, Z., Chen, Z., Li, L., Yang, Y., Li, B., and Yi, J. On the limitations of denoising strategies as adversarial defenses. *CoRR*, abs/2012.09384, 2020. URL https://arxiv.org/abs/2012.09384.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2020.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pp. 506–519, 2017.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *CoRR*, abs/1710.05941, 2017. URL http://arxiv.org/abs/1710.05941.

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

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on mnist. In *Seventh International Conference on Learning Representations (ICLR 2019)*, pp. 1–16, 2019.

Shaham, U., Garritano, J., Yamada, Y., Weinberger, E., Cloninger, A., Cheng, X., Stanton, K., and Kluger, Y. Defending against adversarial images using basis functions transformations. *arXiv preprint arXiv:1803.10840*, 2018.

Sitzmann, V. Awesome Implicit Representations - A curated list of resources on implicit neural representations. 2020. URL https://github.com/vsitzmann/awesome-implicit-representations.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Skorokhodov, I., Ignatyev, S., and Elhoseiny, M. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10753–10764, 2021.

Song, S., Chen, Y., Cheung, N.-M., and Kuo, C.-C. J. Defense against adversarial attacks with saak transform. *arXiv preprint arXiv:1808.01785*, 2018a.

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

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33, 2020.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

Yang, Y., Zhang, G., Katabi, D., and Xu, Z. Me-net: Towards effective adversarial robustness with matrix estimation. In *International Conference on Machine Learning*, pp. 7025–7034. PMLR, 2019.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. *CoRR*, abs/1905.04899, 2019. URL http://arxiv.org/abs/1905.04899.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

A. LINAC Implementation Details

A.1. Implicit Neural Representations

A.1.1. RANDOM NUMBER GENERATION FOR INRS

Our LINAC defence is fully deterministic by design. We used a random 64-bit signed integer as the *private key*, which seeded the state of the pseudo-random number generator in JAX (Frostig et al., 2018; Bradbury et al., 2018). The precise value of the *private key* used to train the defended model evaluated throughout this work was: -2314326399425823309. It was itself selected randomly, by initialising the random number generator of the NumPy library (Harris et al., 2020) with seed 42 and using the first int64 integer.

A.1.2. INPUT AND OUTPUT ENCODINGS

Following Mildenhall et al. (2020) we use a positional encoding of pixel coordinates to a higher dimensional space to better capture higher-frequency information. Each pixel coordinate d is normalised to $[-1, 1]$ and transformed as follows:

$$\gamma(d) = [\sin(2^0 \pi d), \cos(2^0 \pi d), \sin(2^1 \pi d), \cos(2^1 \pi d), \ldots, \sin(2^{F-1} \pi d), \cos(2^{F-1} \pi d)] \tag{5}$$

We used $F = 5$ frequencies in all our experiments and a $L = 5$ hidden layer MLP with $H = 256$ units per layer and ReLU non-linearities. Activations in the middle hidden layer were used for computing the LINAC transform, hence $K = 2$.

As per standard practice for CIFAR-10 classification, pixel colour intensities were scaled to have 0 mean across the training dataset and each colour channel separately. Intensities were then standardised to 1 standard deviation across the training dataset, independently across channels.

A.1.3. FITTING

Fitting the parameters of the implicit neural network was done using Adam (Kingma & Ba, 2015), with default parameters and a learning rate $\mu = 0.001$. We used mini-batches with $M = 32$ random pixels and trained for $N = 10$ epochs. An epoch constitutes a pass through the entire set of pixels in the input image with dimensions $I \times J \times C = 32 \times 32 \times 3$ in random order. The total number of optimisation steps performed was 320. A cosine learning rate decay schedule was used for better convergence, with the minimum value of the multiplier $\alpha = 0.0001$ (Loshchilov & Hutter, 2016).

A.1.4. COMPUTATIONAL AND MEMORY REQUIREMENTS

The LINAC transform's computational complexity scales with the number of pixels $(I \cdot J)$ of the input image and the number of epochs through the pixels (N). It takes $I \cdot J \cdot N$ backward passes through the implicit network Φ to fit its parameters ϕ. LINAC's memory complexity is dominated by the number of parameters of the INR $(|\phi|)$. Empirically, the LINAC transform is itself as expensive as inference with a WideResNet-70-16 model (Zagoruyko & Komodakis, 2016) on CIFAR-10 images.

A.2. Defended Classifiers

Since the proposed input transformation preserves spatial structure, we perform image classification using transformed inputs in an identical manner as with RGB colour images, except for the higher number of channels of transformed inputs. Hence, we employ a standard classification pipeline following (Zagoruyko & Komodakis, 2016), using a WideResNet-70-16 classifier. We reiterate that our proposed transformation changes the number of input channels, but not the spatial dimensions. Hence, small differences between our models and other WideResNet-70-16 results reported in the literature could conceivably appear only due to different numbers of input channels. However, practically this leads to less than a 0.2% increase in the total number of model parameters, limited to the first convolutional layer, which uses filters with 256 channels instead of 3.

We used the Swish activation function proposed by Ramachandran et al. (2017) for all the classifiers. Training was performed with Nesterov Momentum SGD (Tieleman & Hinton, 2012) $m = 0.9$, using mini-batches of size 1024, for a total of 1000 epochs, or 48880 parameter updates. The initial learning rate was $\mu = 0.4$, reduced by a factor of 10 four times, at epochs: 650, 800, 900 and 950. We performed a hyper-parameter sweep over the weight-decay scale with the following grid: $\{0., 0.0001, 0.0005, 0.0010\}$. We maintain an exponential moving average of classifier parameters (with a decay rate of $r = 0.995$); we report accuracies using the final average of classifier parameters.

A.2.1. PERFORMANCE CONSIDERATIONS

We use the CutMix data augmentation strategy of Yun et al. (2019) directly on RGB images from the training set of CIFAR-10, prior to transforming them with LINAC. This has an impact on computational considerations, since pre-computing the transformed dataset offline in order to save training time becomes more challenging. For ease of prototyping we chose to implement LINAC as a preprocessing layer, which could have an impact on training time if used naively, but not if the transformation is applied asynchronously on the buffer of data feeding the device used for model training. We also found empirically that the proposed transformation renders itself to very effective parallelisation using modern SIMD devices, despite the fact that there is no parameter sharing between implicit models of different inputs; this is likely due to the ability of modern libraries such as JAX (Frostig et al., 2018) to vectorise operations across tensors holding parameters for many distinct neural networks.

It is important to note that inference and training costs of defended classifiers are roughly double those of the nominal classifier. Hence, the LINAC transform has comparable cost to inference with a WideResNet-70-16 model.

B. Evaluation Details

B.1. Attacks with Surrogate Models

We provide a breakdown of evaluations using surrogate models initially reported in Section 5. We report the best 10 keys from the brute-force attack on the *private key* in Table 4. These keys were also used to train surrogate models defended with LINAC for use in transfer attacks, see Table 5 for complete results. Reconstruction-based surrogate models defended with modified LINAC, and using the same 10 best-guess attacker keys, were used to perform BPDA transfer attacks, reported in Table 6.

Position	Clean Test Accuracy (%)	Attacker Key
1	57.00	1383227977468296715
2	55.00	-3328443931658504707
3	55.00	-127094507362684985
4	55.00	-7808219206569127925
5	55.00	-8772667224621836765
6	55.00	-706407928311704852
7	54.00	8263151932495004089
8	54.00	-4594861196100637268
9	54.00	-6520968232434877967
10	54.00	-8727766234183220599

Table 4. Top 10 keys in brute-force key attack, also used to train surrogate models.

LINAC Defence		Defended Surrogate Source Models (Attacker Keys)										*Best Adversary*
Norm	**Attack Name**	key_1	key_2	key_3	key_4	key_5	key_6	key_7	key_8	key_9	key_{10}	(against all models)
	AA	93.05	92.99	93.06	93.00	93.01	93.05	93.00	93.00	93.15	93.21	84.00
	MT	89.70	89.48	89.63	89.54	89.59	89.55	89.36	89.58	89.69	89.47	85.70
L_∞	PGD	88.05	88.19	88.20	88.23	88.24	88.17	88.06	88.20	88.29	88.10	87.32
	Square	83.37	83.34	83.60	83.43	83.06	83.38	83.41	83.30	83.36	83.21	75.91
	Best Known	82.22	82.10	82.43	82.26	81.93	82.33	82.24	82.10	82.32	82.17	75.64
	AA	91.20	91.15	91.24	91.17	91.18	91.18	91.20	91.20	91.21	91.16	88.27
	MT	90.29	90.47	90.30	90.50	90.54	90.37	90.20	90.32	90.56	90.25	87.31
L_2	PGD	88.99	89.00	88.92	88.93	88.93	89.08	88.97	88.98	88.95	89.03	88.36
	Square	87.84	87.83	87.77	88.15	87.88	88.11	88.19	87.94	87.85	88.12	84.08
	Best Known	86.51	86.41	86.59	86.42	86.33	86.62	86.50	86.58	86.58	86.53	83.48

Table 5. CIFAR-10 test set robust accuracy (%) of a single LINAC defended classifier w.r.t. a suite of L_∞ and L_2 transfer attacks, valid under our threat model, using surrogate classifiers defended with LINAC, but trained with attacker-chosen keys. The clean accuracy of our defended classifier is 93.08%.

| LINAC Defence | | Reconstruction-Based Surrogate Source Models Using Attacker Keys | | | | | | | | | | Best Adversary |
Norm	Attack Name	key_1	key_2	key_3	key_4	key_5	key_6	key_7	key_8	key_9	key_{10}	(against all models)
L_∞	AA	91.47	91.24	91.21	91.31	91.32	91.35	91.31	91.63	91.33	91.46	59.40
	MT	68.35	69.13	67.71	68.63	67.53	69.07	68.81	67.98	68.40	69.24	55.37
	PGD	69.35	71.57	69.15	70.55	69.30	70.46	69.67	69.48	69.97	70.85	56.00
	Square	82.91	82.98	82.47	82.92	82.66	82.80	82.84	82.91	82.91	82.93	69.14
	Best Known	62.87	64.59	62.63	63.65	62.40	64.02	63.54	62.93	63.40	64.15	51.17
L_2	AA	85.94	86.29	85.95	86.51	85.52	86.16	86.28	86.34	86.46	86.27	74.59
	MT	82.12	82.27	81.91	82.13	81.78	82.42	82.24	82.07	82.21	81.96	74.98
	PGD	82.23	82.77	82.20	82.60	81.89	82.33	82.53	82.38	82.69	82.68	75.00
	Square	87.30	87.34	87.22	87.40	87.35	87.44	87.52	87.21	87.26	87.46	83.26
	Best Known	78.68	79.15	78.36	78.93	78.46	79.08	78.92	78.83	78.80	78.46	71.89

Table 6. CIFAR-10 test set robust accuracy (%) of a single LINAC defended classifier according to a suite of L_∞ and L_2 BPDA attacks, valid under our threat model, using reconstruction-based surrogate classifiers.

B.2. Transfer Attacks with Adversarially Trained Models

For mounting transfer attacks we have taken adversarially trained models from previous work (Rebuffi et al., 2021), with checkpoints available online[1]. These models have been adversarially trained on CIFAR-10 using additional synthetic generated data and CutMix data augmentation. To mount transfer attacks we use the WideResNet-106-16 model (trained to defend against L_∞ norm-bounded perturbations of size $\epsilon = 8/255$) and the WideResNet-70-16 model (trained to defend against L_2 norm-bounded perturbations of size $\epsilon = 0.5$).

B.3. PBA Implementation Details

B.3.1. PBA FOR LINAC

We used a single convolutional layer ($k = 3 \times 3$) with biases to implement $h_\psi(x)$, the PBA of the nuisance transformation, mapping from the 3 RGB channels of input images to the $H = 256$ channels output by LINAC.

The parameters ψ of the bypass approximation were trained by minimising the cross-entropy loss on the CIFAR-10 training set using Momentum SGD with a learning rate $\mu = 0.1$. 100 epochs sufficed to optimise PBA parameters, with four learning rate reductions by a factor of 0.1 at epochs: 65, 80, 90, 95.

B.3.2. PBA FOR BLOCK PIXEL SHUFFLE

We implemented the Block Pixel Shuffle defence of AprilPyone & Kiya (2021a) using blocks of size 4×4, as recommended in the original work. We used the same *private key* value as that of our defended LINAC classifier. The *private key* serves as the seed of a pseudo-random number generator, which is used to sample a permutation of all pixel positions in a block. The same permutation is applied to all blocks. We illustrate the transform in Figure 5.

Figure 5. Example of Pixel Block Shuffle transformation (AprilPyone & Kiya, 2021a). An original CIFAR-10 image (left) is split into a grid of 4×4 blocks of adjacent pixels, and the same random permutation is used to shuffle pixel positions within every block (middle). The transformed image is constructed by spatially concatenating the blocks according to their original positions in the grid (right).

A classifier defended with Block Pixel Shuffle was trained with the same procedure as our defended LINAC classifier. We can report a clean CIFAR-10 test set accuracy of 97.03%, which is higher to that reported by AprilPyone & Kiya (2021a),

---

[1]https://github.com/deepmind/deepmind-research/tree/master/adversarial_robustness

but consistent with the superior CutMix (Yun et al., 2019) data augmentation procedure we used for all defended classifiers.

According its own "white-box" threat model (AprilPyone & Kiya, 2021a), all the implementation details of the defence are known to an attacker except the *private key*. We exploit the block structure and use a single linear layer without biases, and initialised with the identity mapping, to compute a parametric bypass approximation (PBA) for the this defence. We found that using a smaller initial learning rate $\mu = 0.001$ results in stable convergence. We used 300 epochs to optimise PBA parameters, with four learning rate reductions by a factor of 0.1 at epochs: 275, 285, 290, 295.

An extensive evaluation of the resulting defended classifier is given in Table 7. We find that transfer attacks which are agnostic to the defence can be more successful when adversarial examples are computed using robust source models, but one may infer some level of robustness. Using PBA attacks valid under the threat model ("white-box") we successfully circumvent the defence, with a *Best Known* CIFAR-10 robust test-set accuracy of 0% under adversarial perturbations of size up to $\epsilon = 8/255$ in L_∞ norm, and up to $\epsilon = 0.5$ in L_2 norm.

Block Pixel Shuffle Defence		Transfer Attack Source Models			Adaptive Attacks	*Best Adversary*
Norm	Attack Name	Nominal Source	Adversarial Training ($\mathbf{L_\infty}$)	Adversarial Training ($\mathbf{L_2}$)	PBA	All Source Models
L_∞	AA	85.78	69.09	73.86	0.18	0.00
	MT	78.87	56.49	27.72	0.00	0.00
	PGD	69.17	39.05	31.19	0.00	0.00
	Square	69.16	46.25	42.63	0.00	0.00
	Best Known	60.65	30.61	21.17	0.00	0.00
L_2	AA	94.14	90.02	83.35	0.02	0.00
	MT	93.93	92.25	77.02	0.00	0.00
	PGD	92.92	87.54	77.80	0.02	0.02
	Square	91.69	88.80	84.88	6.13	6.13
	Best Known	90.41	85.29	69.00	0.00	0.00

Table 7. CIFAR-10 test set robust accuracy (%) of Block Pixel Shuffle approach (AprilPyone & Kiya, 2021a) against standard L_∞ and L_2 bounded attacks using both transfer and our novel PBA strategy.

C. Sensitivity of LINAC to Hyper-Parameters

We performed sensitivity analyses of LINAC to its hyper-parameters. For efficiency reasons we report robust accuracies according to untargeted PGD attacks with 100 steps and 10 restarts, using an adversarially trained robust model (L_2) (Rebuffi et al., 2021) to generate adversarial perturbations.
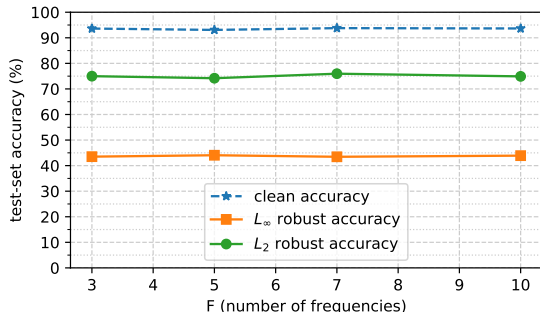

	$F = 3$	$F = 5$	$F = 7$	$F = 10$
Clean Accuracy:	93.61	93.08	93.78	93.65
L_∞ PGD	43.51	44.06	43.46	43.90
L_2 PGD	74.99	74.19	75.96	74.91

Figure 6. CIFAR-10 test-set clean and robust accuracies under transfer attacks of LINAC defended classifiers with different numbers of positional encoding frequencies F, keeping all other hyper-parameters constant.

In Figure 6 we provide a sensitivity analysis across the number of frequencies F used for positional encoding (Mildenhall et al., 2020), keeping all other hyper-parameters the same. Note that we used $F = 5$ for our defended classifier evaluated in

the main paper.
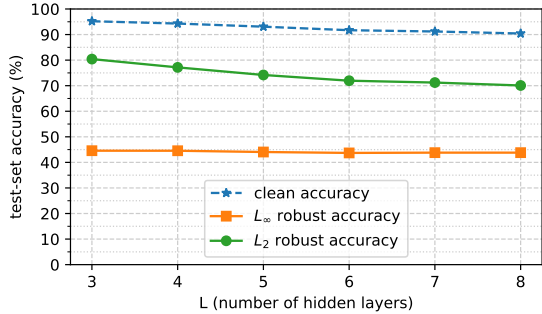
		$L=3$	$L=4$	$L=5$	$L=6$	$L=7$	$L=8$
Clean Accuracy:		95.21	94.30	93.08	91.70	91.20	90.41
L_∞	PGD	44.57	44.55	44.06	43.70	43.81	43.81
L_2	PGD	80.40	77.16	74.19	71.95	71.22	70.08

Figure 7. CIFAR-10 test-set clean and robust accuracies (under transfer attacks) of LINAC defended classifiers with different numbers of implicit network layers L, keeping all other hyper-parameters fixed.

In Figure 7 we vary the number of implicit network layers L, keeping all other hyper-parameters the same, including the representation layer index $K = 2$ and number of epochs $N = 10$. Note that we used $L = 5$ for our defended classifier evaluated in the main paper.
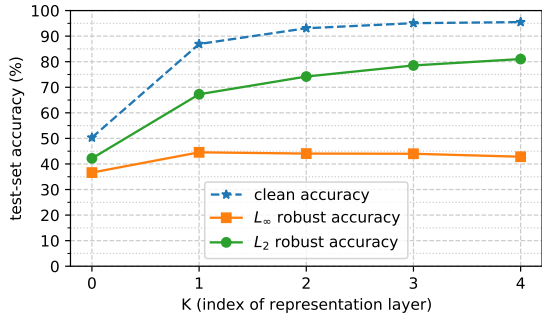
		$K=0$	$K=1$	$K=2$	$K=3$	$K=4$
Clean Accuracy:		50.32	86.92	93.08	95.07	95.49
L_∞	PGD	36.63	44.55	44.06	43.99	42.83
L_2	PGD	42.17	67.26	74.19	78.53	81.03

Figure 8. CIFAR-10 test-set clean and robust accuracies (under transfer attacks) of LINAC defended classifiers with different implicit network layers used to output representations (K), keeping all other hyper-parameters the same.

In Figure 8 we change the index of the LINAC representation layer K, keeping all other hyper-parameters unchanged. Note that we used $K = 2$ for our defended classifier evaluated in the main paper.

In Figure 9 we analyse the sensitivity of LINAC to the number of epochs N, keeping all other hyper-parameters constant. Note that we used $N = 10$ for our defended classifier evaluated in the main paper.

## D. Characterising the LINAC Transform

In Figure 10 we plot learning curves characterising implicit network fitting, as used for our defended classifier. Mean and standard deviation of errors across independent learning processes for the entire CIFAR-10 test-set are plotted as functions of optimisation steps, using a log-scale for errors. The final mean value of such errors is 0.04325, which confirms that our LINAC approach leads to lossy representations.

A histogram of final sum squared errors for the entire test-set of CIFAR-10 is provided in Figure 11.

For a qualitative evaluation of such statistics, we provide examples of original images, their reconstructions and difference images, using LINAC and the *private key* in Figure 12 and, for comparison, a different key in Figure 13. We observe that encoding errors using LINAC are key dependent. Furthermore, significant amounts of information seem to be left out by LINAC. Some difference images could be recognised as the correct class, most likely due to high-frequency information which is not well represented.

Finally, we provide a number of plots for qualitative comparisons of LINAC transforms. Figure 14 shows three different
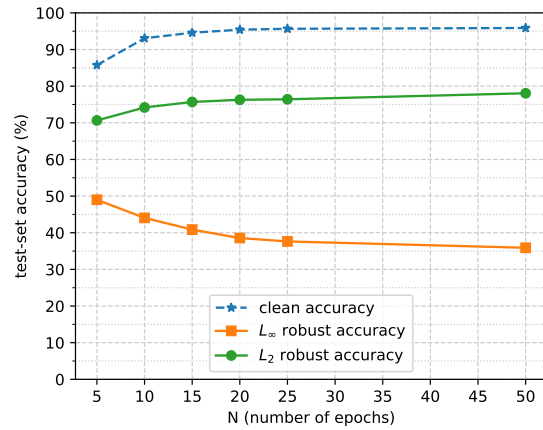
	$N = 5$	$N = 10$	$N = 15$	$N = 20$	$N = 25$	$N = 50$
Clean Accuracy:	85.74	93.08	94.59	95.41	95.66	95.87
L_∞ PGD	49.01	44.06	40.86	38.56	37.63	35.90
L_2 PGD	70.65	74.19	75.69	76.28	76.42	78.06

Figure 9. CIFAR-10 test-set clean and robust accuracies (under transfer attacks) of LINAC defended classifiers with implicit networks trained for N epochs, keeping all other hyper-parameters constant.
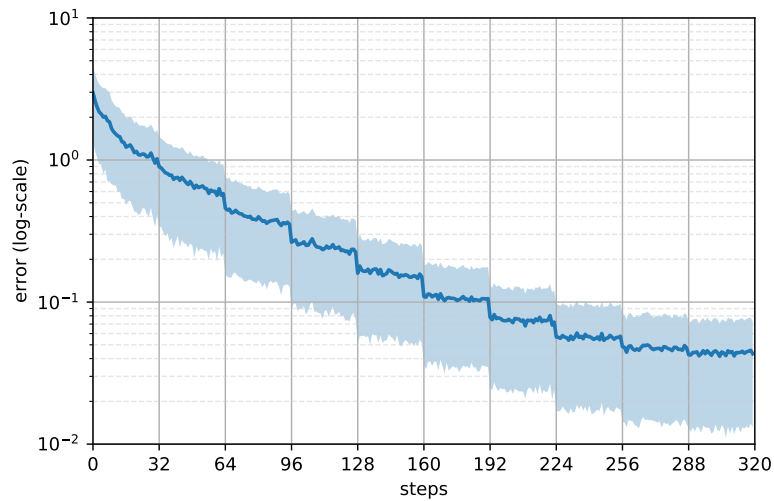
Figure 10. Independent fitting of implicit neural networks to CIFAR-10 test-set images in order to compute their LINAC transforms. Sum squared encoding errors, averaged over pixels, are plotted against fitting steps.

images encoded and their respective LINAC representations encoded as RGB channels. Figures 15, 16 and 17 plot LINAC transforms of the same respective images, but using with different keys, one on each RGB colour channel.
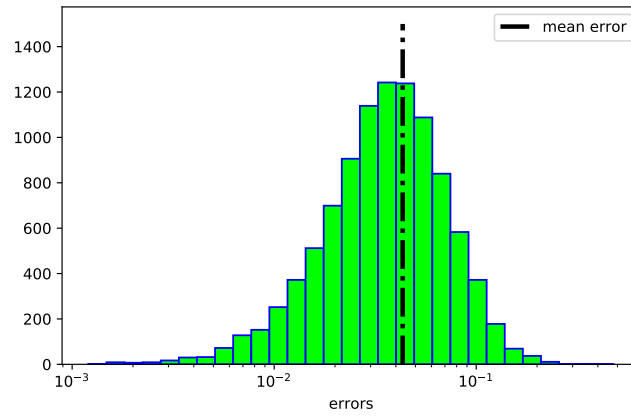
Figure 11. Histogram of LINAC transform encoding errors plotted for the entire CIFAR-10 test-set. The overall mean value of such errors is 0.04325, which confirms that our LINAC approach leads to lossy representations.
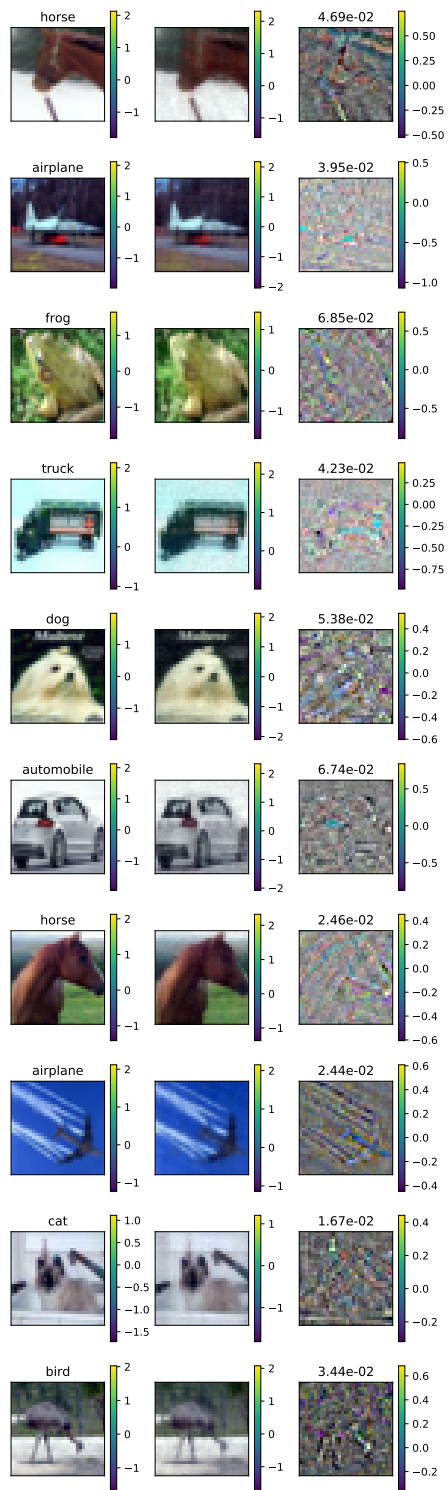
Figure 12. Image approximations computed for LINAC with the *private key*, as used for our defended classifier. Original images and labels are plotted in the first column. Note that labels are not used for LINAC. Implicit network outputs are plotted in the second column. Difference images and sum squared errors, averaged over pixels, are plotted in the third column. Note that LINAC uses lossy image approximations.
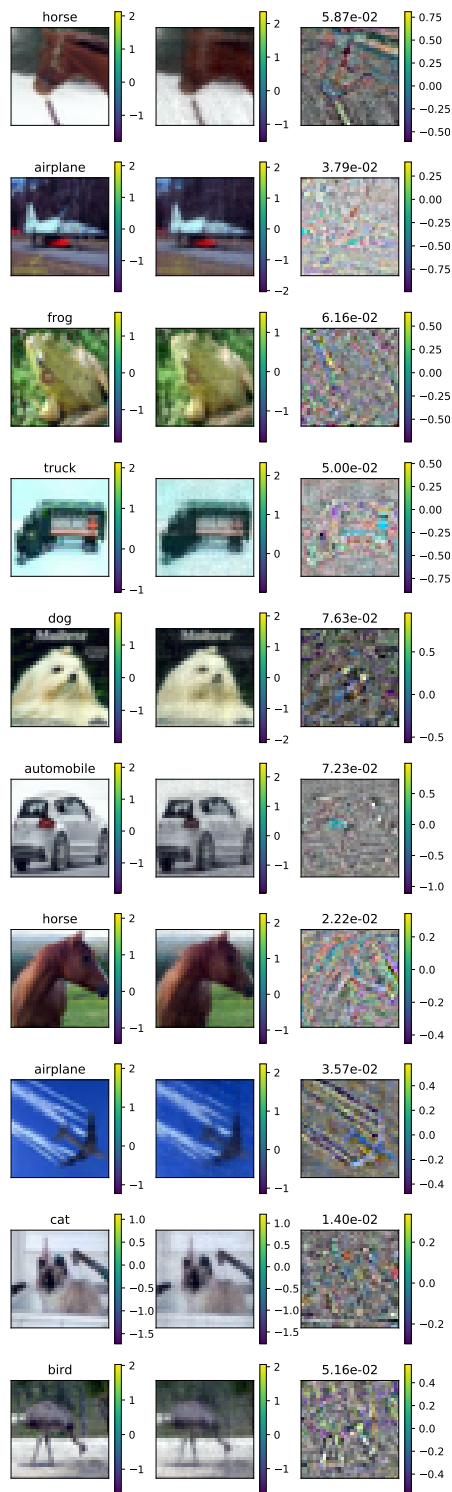
Figure 13. Image approximations computed for LINAC with a **different**, attacker chosen key. Original images and labels are plotted in the first column. Note that labels are not used for LINAC. Implicit network outputs are plotted in the second column. Difference images and sum squared errors, averaged over pixels, are plotted in the third column. Note that LINAC leads to lossy image approximations which are key dependent.
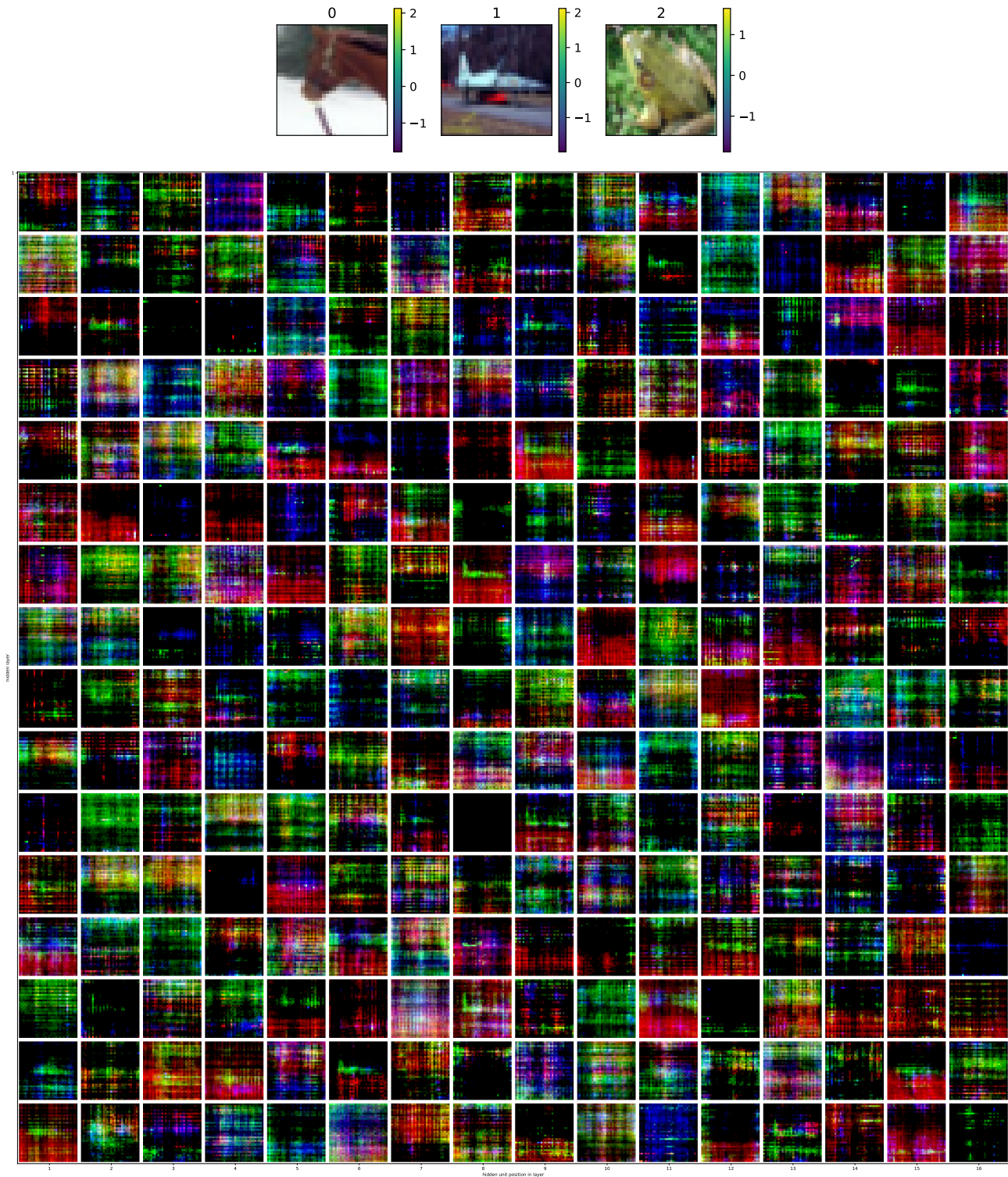
Figure 14. Comparing transforms of the 3 top images using LINAC with the *private key*, as done for our defended classifier. The respective activation images with $H = 256$ channels were plotted in a 16×16 grid of slices of the same size with original images. Respective slices over the channel dimension of activation images were combined as RGB channels in this plot (bottom), in order to compare channel representations for the three input images (top). Each square in the grid represents the activations of a LINAC representation channel for all pixels in the original image. Different values of RGB signify differences in LINAC representations across images.
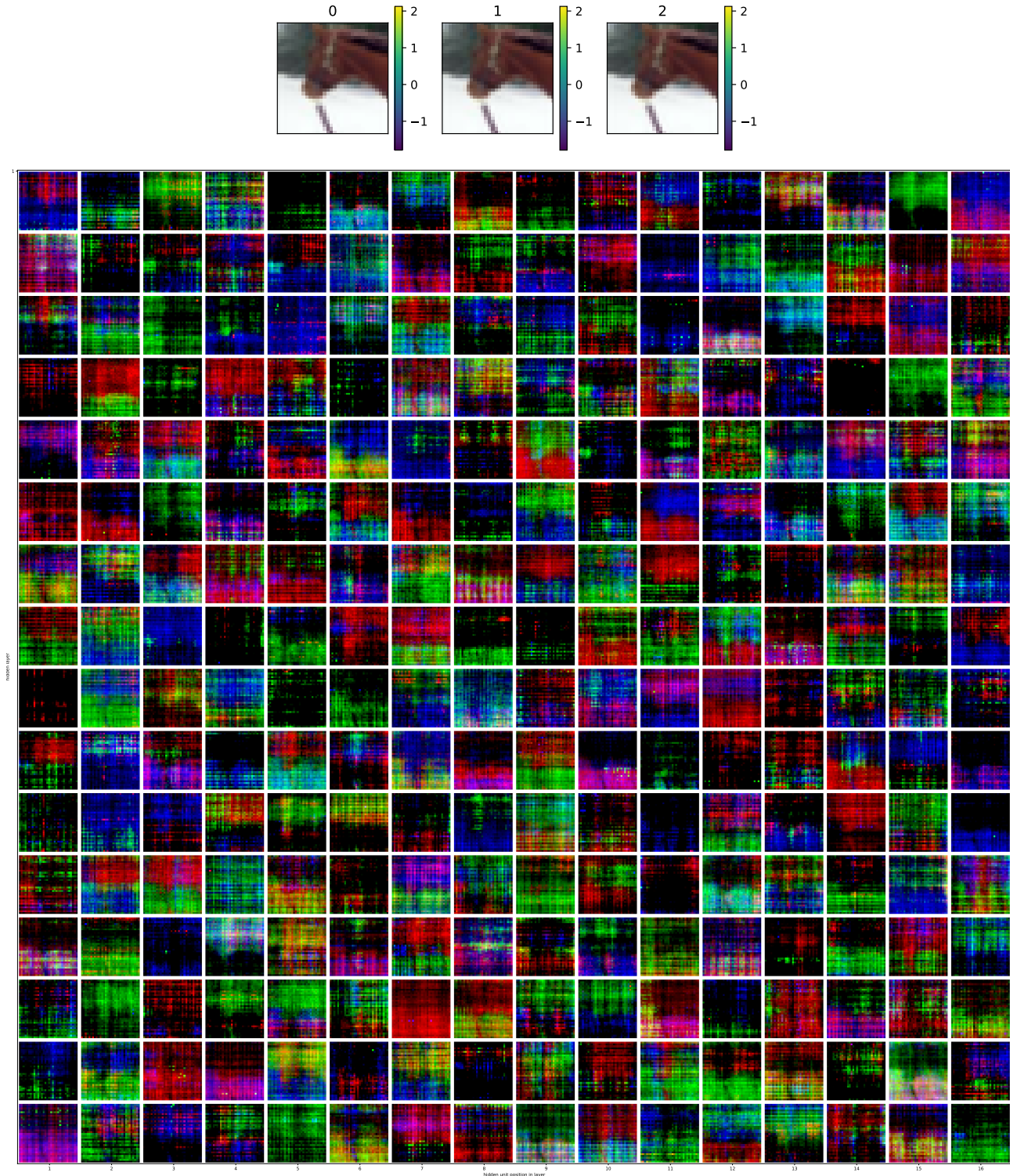
Figure 15. Comparing LINAC transforms of the same image using the *private key* and two other random keys. The respective activation images with $H = 256$ channels were plotted in a 16×16 grid of slices of the same size with original images. Respective slices over the channel dimension of resulting activation images were combined as RGB channels in this plot (bottom), in order to compare channel representations with three different keys for the same input image. Each square in the grid represents the activations of a LINAC representation channel for all pixels in the original image. Different values of RGB signify differences in LINAC representations across keys.
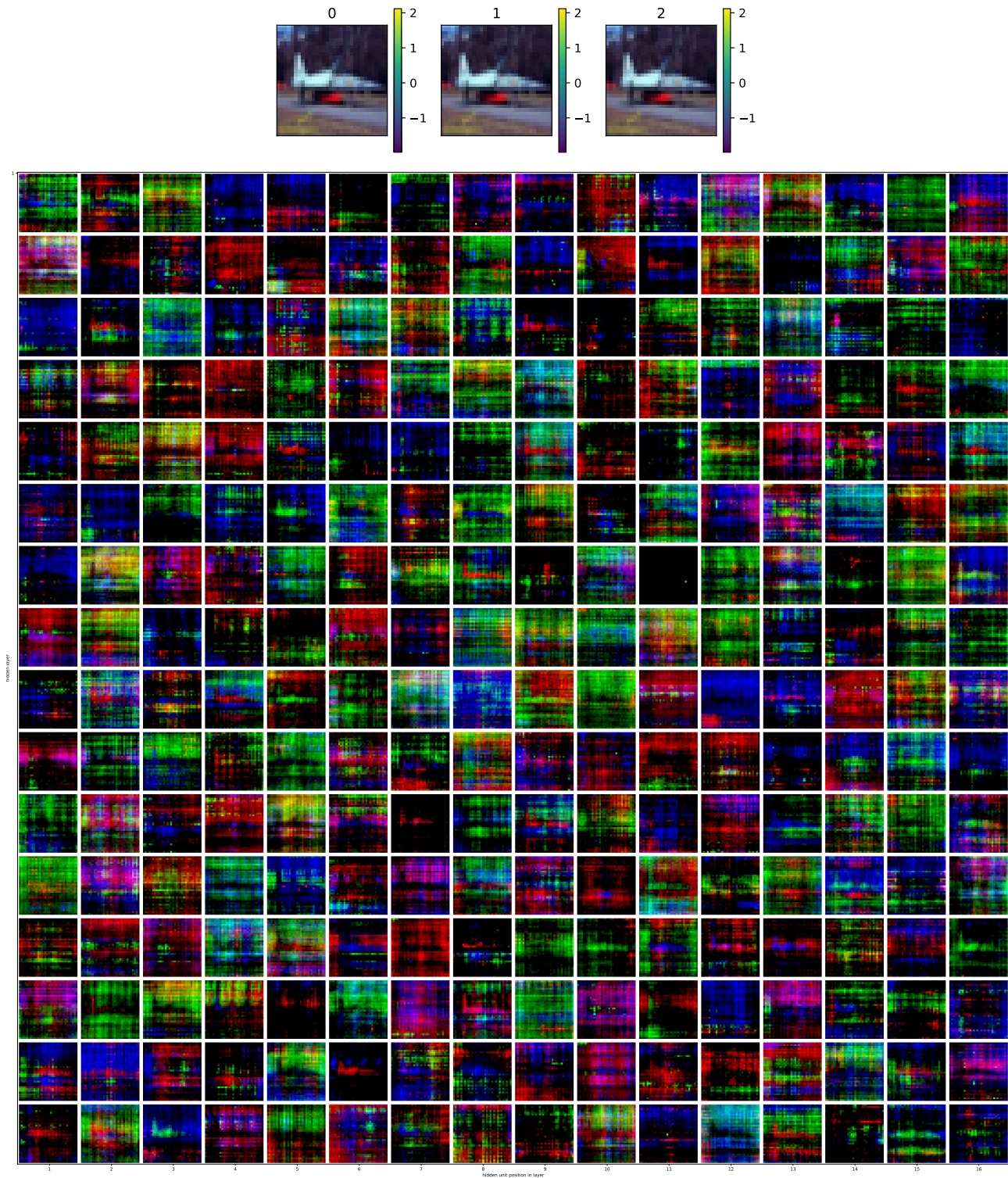
Figure 16. Comparing LINAC transforms of the same image using the *private key* and two other random keys. The respective activation images with $H = 256$ channels were plotted in a 16×16 grid of slices of the same size with original images. Respective slices over the channel dimension of resulting activation images were combined as RGB channels in this plot (bottom), in order to compare channel representations with three different keys for the same input image. Each square in the grid represents the activations of a LINAC representation channel for all pixels in the original image. Different values of RGB signify differences in LINAC representations across keys.
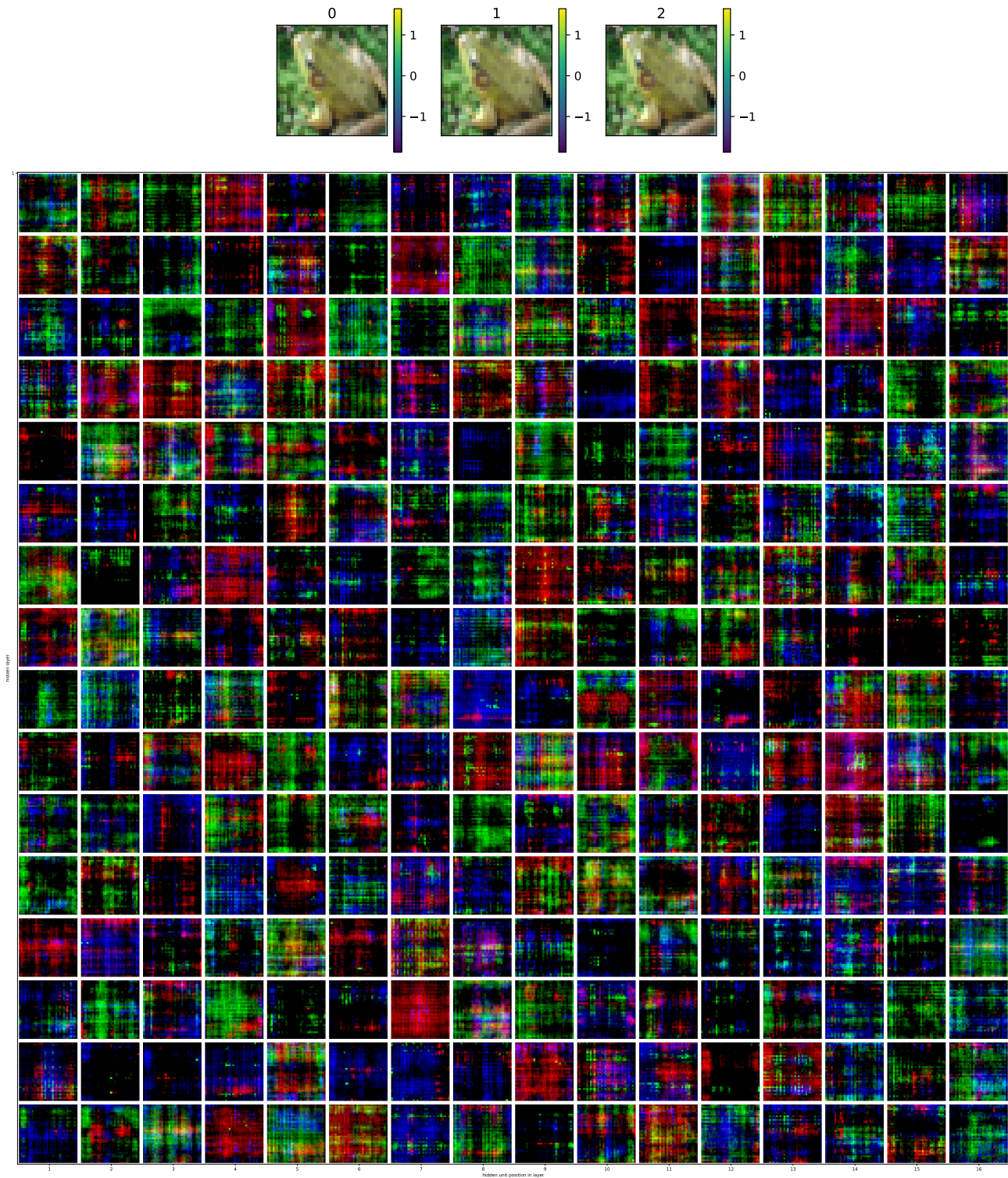
Figure 17. Comparing LINAC transforms of the same image using the *private key* and two other random keys. The respective activation images with $H = 256$ channels were plotted in a 16×16 grid of slices of the same size with original images. Respective slices over the channel dimension of resulting activation images were combined as RGB channels in this plot (bottom), in order to compare channel representations with three different keys for the same input image. Each square in the grid represents the activations of a LINAC representation channel for all pixels in the original image. Different values of RGB signify differences in LINAC representations across keys.