Measuring Representational Robustness of Neural Networks Through Shared Invariances

Vedant Nanda [1 2] Till Speicher [2] Camila Kolling [3] John P. Dickerson [1] Krishna P. Gummadi [2] Adrian Weller [4 5]

Abstract

A major challenge in studying robustness in deep learning is defining the set of "meaningless" perturbations to which a given Neural Network (NN) should be invariant. Most work on robustness implicitly uses a human as the reference model to define such perturbations. Our work offers a new view on robustness by using another reference NN to define the set of perturbations a given NN should be invariant to, thus generalizing the reliance on a reference "human NN" to any NN. This makes measuring robustness equivalent to measuring the extent to which two NNs share invariances. We propose a measure called STIR, which faithfully captures the extent to which two NNs share invariances. STIR repurposes existing representation similarity measures to make them suitable for measuring shared invariances. Using our measure, we are able to gain insights about how shared invariances vary with changes in weight initialization, architecture, loss functions, and training dataset. Our implementation is available at: https://github.com/nvedant07/STIR.

1. Introduction

As deep neural networks are increasingly deployed in real-world scenarios, robustness of their automatically learned feature representations has emerged as a key desideratum. Prior works have defined many measures of robustness corresponding to different types of synthetically-generated or naturally-occurring perturbations that can be applied to the inputs and their distributions (e.g., adversarial inputs (Biggio et al., 2013; Szegedy et al., 2013; Papernot et al., 2016) or distributional shifts (Geirhos et al., 2018; Hendrycks & Dietterich, 2019; Engstrom et al., 2019; Fawzi & Frossard, 2015; Taori et al., 2020)). The common property underlying the different robustness measures is that they all attempt to capture *the extent to which the learned representations remain invariant (i.e., unchanged) under some defined set of perturbations*.

In this paper, we propose a new way to study, understand, and characterize robustness of neural networks. Our key insight is that the set of input perturbations against which a neural network's robustness is measured, can itself be defined by another *reference* neural network. Specifically, given a reference neural network, we first obtain a set of input perturbations that are imperceptible to the reference network (i.e., find inputs with invariant reference representations), and then check the extent to which representations of other neural networks are invariant to these perturbations. Our proposal allows us to measure relative invariance of two neural network representations and estimate the degree to which the two neural networks share representational invariance. Intuitively, our proposal generalizes the often unstated, but implicit assumption behind all interesting sets of perturbations used in robustness studies today: they are perturbations that are imperceptible to a particular reference neural network, the human brain.

Comparing representational invariance of two neural networks is an important aspect of determining their representational (or perceptual) alignment. Assessing representational alignment is crucial for a future society with interacting agents controlled by neural networks (e.g., cars driven by different deep learning systems). Additionally, the ability to measure relative invariance, and therefore robustness, of deep neural network representations can offer insights into interesting questions such as: when updating a model, to what extent are invariances preserved (which may be crucial to regulators for safety assurance)? How does representational invariance vary with the choice of network architectures, loss functions, random weight initialization, and datasets used in the training process?

Our work is inspired by and builds upon previous studies investigating similarity between deep neural network representations (Raghu et al., 2017; Morcos et al., 2018;

[1]University of Maryland, College Park [2]Max Planck Institute for Software Systems [3]PUCRS [4]University of Cambridge [5]The Alan Turing Institute. Correspondence to: Vedant Nanda <vedant@cs.umd.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Kornblith et al., 2019a; Nguyen et al., 2020). However, we find that existing representational similarity literature focuses *narrowly* on comparing two representations of data samples drawn from a specific input distribution, ignoring representations of data samples outside of the distribution or changes in representation caused by input perturbations for which one of the two representations remains invariant. As such, existing measures of similarity between neural network representations offer *no* insight into their robustness and consequently, their alignment. Nevertheless, we retain the compelling (axiomatic) properties of existing similarity measures, by re-purposing them to measure relative invariance. Specifically, for input perturbations imperceptible to the reference neural network, we quantify the invariance in representations of the other neural network using a popular similarity index called Centered Kernel Alignment (CKA) (Kornblith et al., 2019a).

To summarize, our key contributions are as follows:

- We propose a measure of shared invariances between two representations that is based on the models which generated them. We show that our measure faithfully captures shared invariances between two models where existing measures of representation similarity (such as CKA) do not work adequately.

- Our proposal repurposes existing representation similarity measures to measure shared invariance, thus preserving all the (desirable) axiomatic properties of these measures.

- Using our measure we are able to derive novel insights about the impact of weight initialization, architecture, loss and training dataset on the shared invariances between networks. Our initial results show that our measure is a promising evaluation tool to better understand deep learning.

- We find that typically the shared invariance between models reduces for later layers, however when trained using adversarial training, the same models end up with higher shared invariances even in later layers. We also see that models with residual connections tend to have high shared invariances among them than between other non-residual models.

2. Measuring Representational Robustness

Robustness is defined as "perform(ing) without failure under a wide range of conditions" (Merriam-Webster, 2022). Applying the definition in the context of learning models, we can disentangle two distinct requirements for a model to be considered robust: i) the model must produce correct outputs, *i.e.,* have high accuracy, and ii) these outputs must be produced consistently for a diverse set of inputs, *i.e.,* the outputs of the model must be *invariant* to irrelevant perturbations (changes) in the input. Extensive literature on robust learning (Szegedy et al., 2013; Papernot et al., 2016; Goodfellow et al., 2015) suggests that it is quite hard to train models that achieve high accuracy on standard benchmarking datasets and high invariance to irrelevant (adversarially-generated or naturally-occurring) perturbations simultaneously (Tsipras et al., 2018; Madry et al., 2019; Zhang et al., 2019). Reconciling correctness and invariance requirements remains a topic of active research.

Here, we investigate the robustness of neural network representations that are learned in the process of generating outputs. Specifically, we attempt to quantify the invariance of learned neural network representations to *irrelevant* perturbations in the inputs. Intuitively, a high representational invariance is a necessary (though not sufficient) condition for a neural network model to be robust.

2.1. A Relative Invariance Framework

In order to quantify the invariance of a neural network's representations to irrelevant perturbations to inputs, we need to first define the set of irrelevant input perturbations. We begin by assuming the availability of some *reference* neural network model. We then define the set of irrelevant input perturbations as those changes that do not cause any change in the reference model's representation (i.e., perception) of the inputs. Finally, we quantify how invariant the neural network's representations are to these irrelevant input perturbations.

Our framework effectively quantifies invariance of one neural network's representation relative to another reference neural network. Our use of a reference neural network model is inspired by how human perception is used as a reference to determine which adversarial perturbations (Szegedy et al., 2013) or image corruptions (Geirhos et al., 2018; Hendrycks & Dietterich, 2019) or image transformations (Engstrom et al., 2019; Fawzi & Frossard, 2015) do not alter the perception of inputs and are, hence, considered irrelevant perturbations. Next, we provide a more formal description of our framework.

2.2. Problem Setting

Given a reference neural network $m_1 : \mathbb{R}^m \mapsto \mathbb{R}^{d1}$ and n samples ($X \in \mathbb{R}^{n \times m}$) from a given data distribution ($\mathcal{D}$), our goal is to define a measure of how invariant a given target network $m_2 : \mathbb{R}^m \mapsto \mathbb{R}^{d2}$ is to perturbations of samples X that are imperceptible by m_1, *i.e.,* do not change their representations according to m_1.

Invariant input perturbations for a reference model (X') In our framework, the invariances that are desirable are determined with respect to a reference model, m_1. To this end, we introduce the notion of *Identically Represented Inputs (IRIs)*. For any given input data point x and a given reference model m_1, IRIs is the set of all data points that are mapped to the same representation as x by m_1. Formally,

$$\text{IRI}_{\text{strict}}(x; m_1) = \{x' \mid m_1(x') = m_1(x)\} \quad (1)$$

In other words, m_1 is *invariant* to and cannot perceive any difference between x and any $x' \in \text{IRI}(x; m_1)$. In practice, exact equality is hard to achieve, and thus we relax this formulation so that x and x' are *almost* indistinguishable for m_1, *i.e.,*, for a small enough δ,

$$\text{IRI}_{\text{relax}}(x; m_1) = \{x' \mid \frac{||m_1(x') - m_1(x)||_2}{||m_1(x)||_2} \leq \delta\} \quad (2)$$

Going forward we use the relaxed formulation of IRIs and thus omit the subscript. For each of the n input points $x \in X$, if we pick a x' from $\text{IRI}(x; m_1)$, we get a corresponding batch of n samples X' such that $m_1(X') \approx m_1(X)$.

This however raises a key question: **how to sample x' from an infinitely large set IRI$(x; m_1)$?** We argue that there are two key ways to do this: *arbitrarily* or *adversarially*. We can randomly choose a sample from $\text{IRI}(x; m_1)$, in which case we get *arbitrary* IRIs, or we can pick x' adversarially with respect to m_2, such that the representations $m_2(x')$ and $m_2(x)$ are farthest apart, in which case we get *adversarial* IRIs. We discuss this in more detail in section 2.3 and propose two invariance measures corresponding to these two choices in section 3.1. We also show in Table 1 that takeaways about shared invariance can vary greatly depending on the choice of arbitrary or adversarial IRIs.

Measuring invariance of the target model on (X, X') Once we obtain X and X', our key idea is to capture the extent to which m_1 and m_2 share invarances, by measuring the degree to which representations assigned by m_2 to X and X' are *similar*. Specifically, we want to quantify the degree of similarity between two sets of n data representations $Y = m_2(X) \in \mathbb{R}^{n \times d2}$ and $Z = m_2(X') \in \mathbb{R}^{n \times d2}$. To this end, we can make use of any of the existing representation similarity measures (S_r), such as CKA (Kornblith et al., 2019a), and variants of CCA (Morcos et al., 2018; Raghu et al., 2017) as they're designed specifically to measure similarity between two sets of representations. This yields a *shared invariance measure, S_i*, which can now be formally defined as:

$$\text{S}_\text{i}(m_2 | m_1, X, X', \text{S}_\text{r}) = \text{S}_\text{r}(m_2(X), m_2(X')) \quad (3)$$

Note that while the traditional use of S_r, *i.e.,*, $\text{S}_\text{r}(m_1(X), m_2(X))$ does not measure shared invariance between m_1 and m_2 (we give concrete arguments for the same in Section 3.2), our proposed measure (Eq 3) shows how existing S_r measures can be repurposed to measure shared invariance.

It's important to note that our measure is a directional one since IRIs are defined relative to the reference model m_1. Other than the reference and target models m_1 and m_2, our measure takes given input points X and a representation similarity measure (S_r) as inputs. We discuss the concrete instantiations of S_i used in this work in Section 3.1. First, however, we describe how to construct X' given X.

2.3. Generating IRIs

Arbitrary IRI Operationalizing S_i as defined in Eq 3 requires sampling X' for a given X. Here, we provide a simple way to empirically estimate X' given X (wrt. m_1).

We leverage the key insight that m_1 can map multiple different inputs to the same representation (since m_1 is a highly non-linear deep neural network) which can be found using representation inversion (Mahendran & Vedaldi, 2014). For a given set of inputs typically drawn from the training distribution $X = [x_1 ... x_n]$ and a reference model m_1, we generate $X' = [x'_1 ... x'_n]$ that are all mapped to similar representations as X by m_1, *i.e.,* $m_1(X) \approx m_1(X')$. Note that X and X', are, by construction IRIs. This is achieved by performing the following optimization:

$$\text{argmin}_{X'} \mathcal{L}(X'), \quad (4)$$

$$\text{where } \mathcal{L}(X') = ||m_1(X') - m_1(X)||_2.$$

This can be approximated using gradient descent by repeatedly performing the following update (where α is the step size):

$$X' = X' + \alpha * \nabla_{X'} \mathcal{L}. \quad (5)$$

A key consideration in solving this optimization is that we must start with some initial value of X', *i.e.,* we must choose a seed X' from which to start the gradient descent. Different seeds can lead to different solutions of X'. We find that in practice randomly picked seeds give fairly stable estimates[1]. Since this scheme only optimizes for similar representation of x and x' on m_1 ($\forall x \in X$) and starts from a random initial value of x', it simulates a random sample from the (possibly infinitely) large set $\text{IRI}(x; m_1)$.

Adversarial IRIs We can alter the way we find X' such that the resulting (X, X') are still, by definition IRIs but are optimized to generate very distinct outputs on m_2, the model for which we're measuring shared invariance. This can be achieved by using exactly the same procedure as in

[1]Since we deal with images we sample each pixel value as a random integer from $0 - 255$ with uniform probabilities.

Eqs 4 and 5, except we now change $\mathcal{L}$ to:

$$\mathcal{L}_{adv} = ||m_1(X') - m_1(X)||_2 - \\ ||m_2(X') - m_2(X)||_2.$$

Solving for X' using $\mathcal{L}_{adv}$ ensures that the inputs are still similarly represented as X on m_1 and thus (X, X') are IRIs. However this ensures that any measure of S_i on such IRIs will be a worst case estimate. Such IRIs have been referred to as *controversial stimuli* (Golan et al., 2020) in existing literature.

3. Measuring Shared Invariances

3.1. STIR, an instantiation of $\mathrm{S_i}$

Using Eqs 4&5 we can generate k different (X, X'), by repeatedly sampling X k times from a given distribution (typically the training distribution of m_1, *e.g.,* the train or test set). Now, we define $\text{STIR}(m_2|m_1, X, \mathrm{S_r})$ as follows:

$$\text{STIR}(m_2|m_1, X, \mathrm{S_r}) = \frac{1}{k}\sum_{X'} \mathrm{S_r}(m_2(X), m_2(X')). \tag{6}$$

Here, we find X' using representation inversion as described in Section 2.3. We call this measure **S**imilarity **T**hrough **I**nverted **R**epresentations — STIR.

When all X' are chosen in an adversarial manner (*i.e.,*using $\mathcal{L}_{adv}$ as described in Section 2.3), we can estimate the worst case STIR as:

$$\text{STIR}_{adv}(m_2|m_1, X, \mathrm{S_r}) = \frac{1}{k}\sum_{X'_{adv}} \mathrm{S_r}(m_2(X), m_2(X')). \tag{7}$$

Both Equations 6&7 are parametrized by S_r. For our purpose we use linear Centered Kernel Alignment (CKA) (Kornblith et al., 2019a) as the similarity measure, *i.e.,*$S_r =$ Linear CKA. CKA has been adopted by the community as the standard measure for representation similarity and has been used by many subsequent works to derive important insights about deep learning (Nguyen et al., 2020; Raghu et al., 2021; Neyshabur et al., 2020). CKA also has certain desirable axiomatic properties such as invariance to isotropic scaling and orthogonal transformations, which other methods such as SVCCA (Raghu et al., 2017) and PWCCA (Morcos et al., 2018) do not possess. Importantly, CKA is *not* invariant to every invertible linear transformation, which is not true of SVCCA or PWCCA. We refer to the paper (Kornblith et al., 2019a) for an extensive discussion on why these are desirable properties for any similarity measure. While CKA, as proposed, can work with any kernel function, results show that Linear CKA works just as well as RBF CKA, so for simplicity we use Linear CKA for all our experiments. Definitions of the similarity metrics listed above can be found in Appendix A.

3.2. Why Existing Representation Similarity (S_r) Measures Cannot Measure Shared Invariance

While at first glance a representation similarity measure (RSM) such as CKA (Kornblith et al., 2019a) or one of the others mentioned in section 3.1 might look appealing to measure shared invariance, these measures are in fact not suitable for this task, for several reasons.

First, **current RSMs are not designed for capturing invariance.** Their goal is to measure the degree of correlation between sets of points generated by two different models, that are transformed to be as aligned as possible under certain constraints. Measuring invariance, however, requires capturing the degree of difference between points generated *by the same model*, which should be the same.

Second, **RSMs don't interact with the model.** All existing RSMs are evaluated in a two-step process, by first obtaining collections of representations $Y = m_1(X), Z = m_2(X)$ from two models and then computing similarity based on those representations as $\mathrm{S_r}(Y, Z)$, without using the models further. From a causal perspective, an invariance is an intervention on a model's input that does not lead to a change in the model's output. However, a central result in the causality literature is that the effect of interventions cannot be determined from observational data alone (Pearl, 2009). Therefore, any metric that aims to make meaningful statements about model invariance needs to interact with the model to make interventions. STIR does this via the process of representation-inversion, however, existing RSMs are purely observational and therefore cannot properly determine invariances.

Third, **sharing invariance between two models is directional.** If model m_1 is constant, it will share all of model m_2's invariances, but not vice versa if m_2 is not constant itself. Existing RSMs are not directional and therefore cannot express these relationships.

3.3. STIR Faithfully Measures Shared Invariance

Training two models (m_1 and m_2) with different random initializations (holding all other things like architecture, loss and other hyperparameters constant), leads to very "similar" representations on the penultimate layer, as measured by instantiations of S_r such as CKA (Kornblith et al., 2019a). We consider two variants of this experiment, where we train two ResNet18 models (on CIFAR10) from different random initializations (keeping everything else same) with 1.) the standard crossentropy loss ($\mathrm{m^{(Vanilla)}}_1$, $\mathrm{m^{(Vanilla)}}_2$), and

	ResNet18 Vanilla (m_1), ResNet18 Vanilla (m_2)		ResNet18 AT (m_1), ResNet18 AT (m_2)		ResNet18 AT (m_1), ResNet18 Vanilla (m_2)	
	$m_1 \| m_2$	$m_2 \| m_1$	$m_1 \| m_2$	$m_2 \| m_1$	$m_1 \| m_2$	$m_2 \| m_1$
STIR	$0.605_{\pm 0.013}$	$0.562_{\pm 0.023}$	$0.934_{\pm 0.003}$	$0.939_{\pm 0.002}$	$0.405_{\pm 0.020}$	$0.509_{\pm 0.011}$
STIR$_{adv}$	$0.085_{\pm 0.004}$	$0.064_{\pm 0.004}$	$0.096_{\pm 0.007}$	$0.078_{\pm 0.005}$	$0.054_{\pm 0.004}$	$0.070_{\pm 0.004}$
CKA	$0.967_{\pm 0.000}$		$0.937_{\pm 0.000}$		$0.536_{\pm 0.000}$	
Acc($m_1(X')$, $m_2(X')$)	$0.521_{\pm 0.061}$	$0.347_{\pm 0.029}$	$0.891_{\pm 0.007}$	$0.901_{\pm 0.002}$	$0.140_{\pm 0.028}$	$0.555_{\pm 0.012}$

Table 1. **[STIR faithfully estimates shared invariance]** Here the two ResNet18s in each column are trained on CIFAR10 with different random initializations, holding every other hyperparameter constant. 1.) For two such models trained using the vanilla crossentropy loss (left), interestingly, we find that STIR highlights a lack of shared invariance, whereas CKA overestimates this value; 2.) when both models are trained using adversarial training (Madry et al., 2019) (middle) STIR faithfully estimates high shared invariance; 3.) Finally STIR is able show how having a directional measure can bring out the differences when comparing a model trained with vanilla loss and adv training (right), whereas CKA being unidirectional cannot derive these insights. All numbers are computed over 1000 random samples from CIFAR10 training set and averaged over 5 runs.

2.) with adversarial training ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{AT})}{}_2$) [2]. Throughout the paper STIR is measured over CIFAR10 training samples with S_r = Linear CKA. Thus, to simplify the notation, we use STIR$(m_1|m_2)$ to mean STIR$(m_1|m_2, X \sim \mathrm{CIFAR10}, \mathrm{LinearCKA})$.

STIR provides insights into shared invariance where CKA fails. Both ($\mathrm{m}^{(\mathrm{Vanilla})}{}_1$, $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$) and ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{AT})}{}_2$) achieve a high similarity score (as measured by CKA, on the penultimate layer of both models). However, we find that such a similarity measurement would be an overestimation of shared invariances between $\mathrm{m}^{(\mathrm{Vanilla})}{}_1$ and $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$ which are much lower when measured as STIR$(\mathrm{m}^{(\mathrm{Vanilla})}{}_1 \,|\, \mathrm{m}^{(\mathrm{Vanilla})}{}_2)$ and STIR$(\mathrm{m}^{(\mathrm{Vanilla})}{}_2 \,|\, \mathrm{m}^{(\mathrm{Vanilla})}{}_1)$, as shown in Table 1. Two models trained using adversarial training ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{AT})}{}_2$) should intuitively have more shared invariances since these models were explicitly trained to be invariant to ℓ_2 perturbations. Indeed, we see that these two models have much higher values of STIR$(\mathrm{m}^{(\mathrm{AT})}{}_1 \,|\, \mathrm{m}^{(\mathrm{AT})}{}_2)$ and STIR$(\mathrm{m}^{(\mathrm{AT})}{}_2 \,|\, \mathrm{m}^{(\mathrm{AT})}{}_1)$.

Sanity Checks for STIR For high values of STIR, intuitively we would expect the representations of the model we're evaluating (m_2) to have similar representations on IRIs, *i.e.,* $m_2(X) \approx m_2(X')$. Since IRIs, by construction, have similar representation on the reference model (m_1), we expect the predictions of m_2 on X' (pred($m_2(X')$)) to agree with predictions of m_1 on X' (pred($m_1(X')$)). Similarly, for low STIR values, we'd expect less agreement between pred($m_1(X')$) and pred($m_2(X')$). We see that this relationship holds as lower STIR values for $\mathrm{m}^{(\mathrm{Vanilla})}{}_1$ and $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$ also correspond to less agreement in their predictions on X' and higher STIR values for $\mathrm{m}^{(\mathrm{AT})}{}_1$ and $\mathrm{m}^{(\mathrm{AT})}{}_2$ correspond to higher agreement in their predictions (as shown in the right of each row of Table 1). To further corroborate that the estimate of shared invariance given by STIR is justified in being lower for ($\mathrm{m}^{(\mathrm{Vanilla})}{}_1$, $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$) than ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{AT})}{}_2$), we generate *controversial stimuli* (Golan et al., 2020) for both pairs of models. Details of how to generate these can be found in the Appendix C. We find that indeed it's significantly easier to generate controversial stimuli for ($\mathrm{m}^{(\mathrm{Vanilla})}{}_1$, $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$) than for ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{AT})}{}_2$) (see Appendix C for the results).

STIR$_{adv}$ shows that in the worst case, there are almost no shared invariances. When measuring shared invariance in the worst case (using STIR$_{adv}$, Eq 7), we see that even in the case of $\mathrm{m}^{(\mathrm{AT})}{}_1$ and $\mathrm{m}^{(\mathrm{AT})}{}_2$ the shared invariance drops close to 0. This is shown in the second row of Table 1. Thus, for the rest of the paper we focus on STIR measured using arbitrary IRIs, since STIR$_{adv}$ gives a very pessimistic estimate of shared invariance and thus is not useful in comparing models.

STIR brings out nuance through directionality. When comparing across training types, *e.g.,* ($\mathrm{m}^{(\mathrm{AT})}{}_1$, $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$), we find that directionality of STIR is able to show that invariances of $\mathrm{m}^{(\mathrm{Vanilla})}{}_2$ are not well captured by $\mathrm{m}^{(\mathrm{AT})}{}_1$ (indicated by the low value of STIR$(\mathrm{m}^{(\mathrm{Vanilla})}{}_1 \,|\, \mathrm{m}^{(\mathrm{AT})}{}_2)$). However, STIR$(\mathrm{m}^{(\mathrm{AT})}{}_2 \,|\, \mathrm{m}^{(\mathrm{Vanilla})}{}_1)$ is much higher. We posit that models trained using AT have a "superior" set of invariances and do not posses "bad" invariances that models trained using the vanila loss exhibit. Thus, when evaluating IRIs from a Vanilla model on a model trained using AT – one can expect lower shared invariance than in the other direction. STIR is able to capture these nuances in comparison between models because of its directionality. Measures of representation similarity (S_r) like CKA do not offer any such insights, since they're not directional.

[2]trained using ℓ_2 threat model with $\epsilon = 1.0$, see (Madry et al., 2019) for more details

	AT IT=10	AT IT=1	VANILLA
AT IT=10 ROB: $51.5_{\pm 0.6}$ CLEAN: $80.4_{\pm 0.4}$	—	$0.93_{\pm 0.01}$ (0.89 ± 0.02)	$0.51_{\pm 0.01}$ (0.56 ± 0.01)
AT IT=1 ROB: $34.6_{\pm 0.6}$ CLEAN: $87.2_{\pm 2.1}$	$0.86_{\pm 0.02}$ (0.89 ± 0.02)	—	$0.52_{\pm 0.01}$ (0.64 ± 0.00)
VANILLA ROB: $0.0_{\pm 0.0}$ CLEAN: $95.0_{\pm 0.1}$	$0.41_{\pm 0.02}$ (0.56 ± 0.01)	$0.34_{\pm 0.05}$ (0.64 ± 0.00)	—

Table 2. ITERS is the number of iterations in the inner loop of Adversarial Training (AT). Each cell shows STIR$(m_j|m_i)$ where m_j is the model on the column and m_i is the model on the row. CKA numbers are shown in (). The three models have robust accuracies (measured under $\ell_2, \epsilon = 1$) such that $AT, IT = 10 > AT, IT = 1 >$ Vanilla.

4. Using STIR to Analyze Model Updates

One motivation for a shared invariance measure like STIR, as mentioned in Section 1, is to monitor how different models "align" with each other. This can then be used to analyze if updates to a model leads to preservation of invariances learned before the update. We first show that STIR can capture relative differences between model invariances and then use STIR to analyze a simulated model update scenario where we incrementally add more training data.

4.1. STIR Captures Relative Robustness

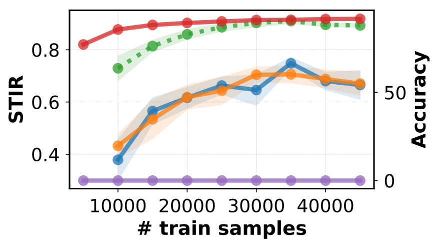
We demonstrate that STIR can capture differences between models of varying degrees of robustness. We know that increasing adversarial robustness increases invariance to ℓ_p ball perturbations. Thus, by construction, if a model m_1 has higher adversarial robustness than another model m_2, then invariances of m_1 should be "superior" to those of m_2 and hence we should see STIR$(m_2|m_1) >$ STIR$(m_1|m_2)$. To empirically test this, we construct three models with varying degrees of adversarial robustness: a model trained using the vanilla crossentropy loss (0%), a model trained using adversarial training (AT) with the usual 10 iterations used to solve the inner maximization of AT (Madry et al., 2019) (51.5%), and a model trained using AT but with only 1 iteration in the inner maximization loop, which gives adversarial robustness somewhere in the middle (34.6%). Table 2 shows comparison between these three models and confirms that STIR$(m_2|m_1) >$ STIR$(m_1|m_2)$ if adversarial robustness $m_1 > m_2$ (measured here by robust accuracy).

4.2. Updating Models With More Data

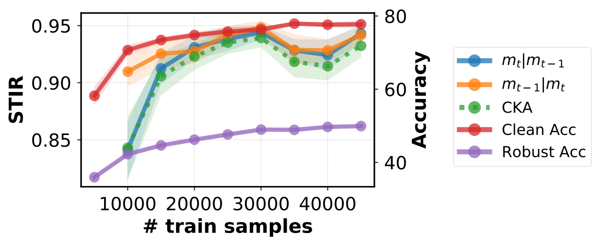
Models deployed in the real world are continuously updated with more data. In such cases, it may be crucial to understand how a model update at a given timestep shares invariance with the model at the previous timestep. To simulate such a scenario, we train a ResNet18 on CIFAR10 where at each timestep, we add $5k$ training samples, *i.e.*,, at timesteps $t = 0, 1, ..., T$, we train the model on $5k, 10k, ..., 45k$ samples (we keep $5k$ samples for a holdout validation set). At each timestep we train for 100 epochs. Figure 1 shows how STIR$(m_t|m_{t-1})$ (and STIR$(m_{t-1}|m_t)$) changes as we progressively add more data. Here m_t is the model at a given timestep and m_{t-1} is the model on the previous timestep. For both AT and Vanilla loss we see STIR$(m_t|m_{t-1})$ and STIR$(m_{t-1}|m_t)$ increase as we add more data. We also see that after a certain amount of data the STIR scores plateau – thus indicating that adding more data has diminishing returns for shared invariance.

5. Evaluating the Impact of Design Choices on Shared Invariance using STIR

A lot of research effort been dedicated towards finding architectures, training schemes and datasets that produce more correct (*i.e.,* accurate) models. However, the effect that these design choices have on the relative invariance of models is still not properly understood. In this section, we leverage STIR to investigate the effect that the various choices in the training pipeline have on shared invariances between models. All evaluations of STIR in this section are performed using the CIFAR10 dataset. See Appendix B for additional details.

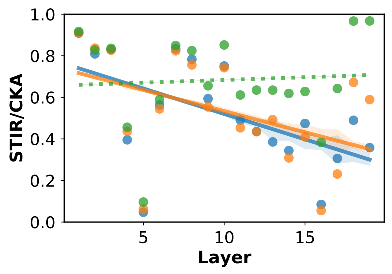


(a) Standard Training

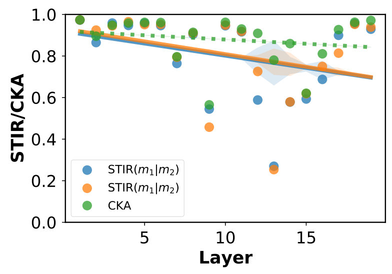


(b) Adv Training

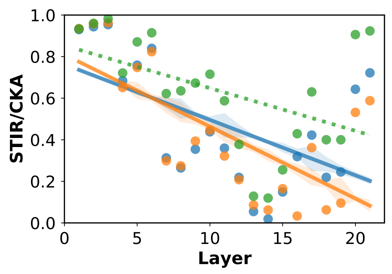
Figure 1. Adding more training data for (x-axis) leads to a monotonic increase in STIR between model at a given timestep (m_t) and the previous timestep (m_{t-1}) (in both directions).



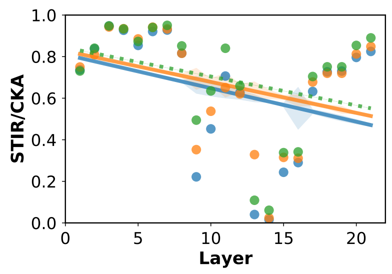
(a) 2 ResNet18 with different init, Vanilla



(b) 2 ResNet18 with different init, AT



(c) 2 VGG16 with different init, Vanilla



(d) 2 VGG16 with different init, AT

Figure 2. **[Role of Random Initialization; CIFAR10]** For models trained using the Vanilla crossentropy loss, we find that only initial layers have high shared invariances. However, when the training procedure explicitly introduces invariances (*e.g.,*adversarial training), then all layers converge to having similarly high shared invariances. Similar trends also hold for deeper variants of ResNet and VGG (results in Appendix D).

5.1. Role of Different Random Initialization, Different Training Datasets

Random Initializations (Kornblith et al., 2019b) find that the same architecture trained from two different random initializations should converge to highly similar representations, *i.e.,*layer k in both models has high similarity. We find that this is not necessarily the case for shared invariances. Fig. 2 shows results for two ResNet18 (different random initialization) and two VGG16 (different random initialization) models trained on CIFAR10 using the vanilla crossentropy loss and adversarial training. We see that models trained with vanilla loss (Fig. 2a & 2c), later layers have lower shared invariances than initial layers, indicated by a negative slope of lines of best fit. However, with adversarial training, both initial and final layers converge to (almost) similarly high levels of shared invariances (indicated by flatter lines of fit in Fig. 2b & 2d). Thus, we conclude that when training from different random initializations, training procedures that explicitly introduce invariance (such as adversarial training) make each layer of two differently initialized models converge to similar shared invariances.

Datasets For the same architecture (ResNet18) trained on different datasets (CIFAR10 and CIFAR100), we evaluate the shared invariances between each of the corresponding layers. We find that in general, initial layers tend to have higher shared invariances, as indicated by the negative slope of the line of best fit in Fig. 3. Interestingly, (Kornblith et al., 2019a) also had a similar observation when measuring similarity for models trained on different datasets. Additionally, we see that shared invariance increases substantially (for all layers) when these models are trained using adversarial training, similar to the random initilaization case, even though the training here is performed on different datasets. We see similar trends for other architectures too (results in Appendix D).

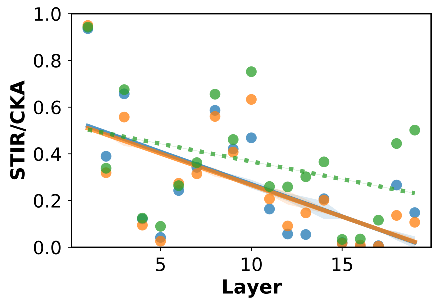
5.2. Different Architectures, Penultimate Layer

We train ResNet18, ResNet34, VGG16 and VGG19 with two different random seeds and using both the vanilla loss and adversarial training (AT). We then compute the shared invariances across all these configurations of models (for the penultimate layer of each model) as shown in Fig. 4.

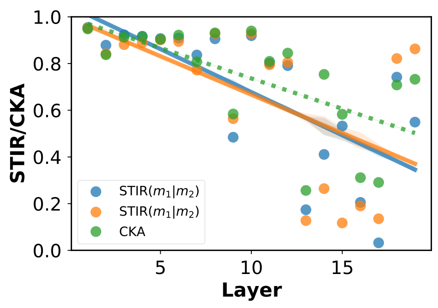
We find that in general, architectures with residual connections (ResNet18 and ResNet34) have high shared invariances amongst themselves, as indicated by high values of STIR amongst ResNets (for both vanilla and AT).

5.3. Different Adversarial Training Methods

As observed in Fig. 4, models trained with AT generally have higher values of STIR amongst them than models trained using the vanilla loss. This raises a natural question: does high STIR between models hold for any kind of training that makes models robust to ℓ_p ball perturbations?

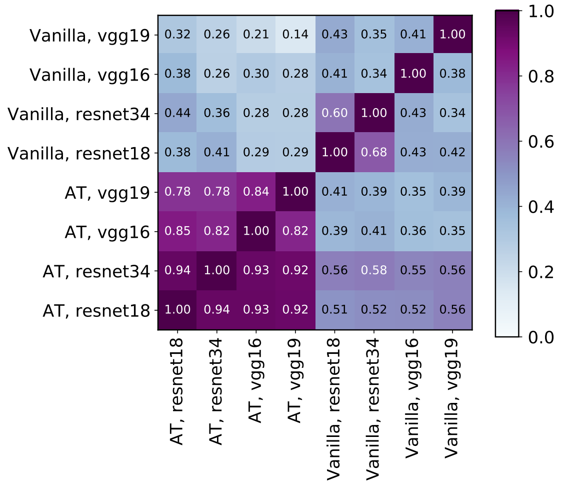


(a) m_1= ResNet18 on CIFAR10 w Vanilla Loss
m_2= ResNet18 on CIFAR100 w Vanilla Loss



(b) m_1= ResNet18 on CIFAR10 w AT
m_2= ResNet18 on CIFAR100 w AT

Figure 3. **[Different Datasets; ResNet18 on CIFAR10 and CIFAR100]** Similar to the finding of (Kornblith et al., 2019a) we find that across datasets, initial layers tend to have more shared invariances. However, we also find that shared invariances increases substantially for all layers with adversarial training (AT).



	AT, resnet18	AT, resnet34	AT, vgg16	AT, vgg19	Vanilla, resnet18	Vanilla, resnet34	Vanilla, vgg16	Vanilla, vgg19
Vanilla, vgg19	0.32	0.26	0.21	0.14	0.43	0.35	0.41	1.00
Vanilla, vgg16	0.38	0.26	0.30	0.28	0.41	0.34	1.00	0.38
Vanilla, resnet34	0.44	0.36	0.28	0.28	0.60	1.00	0.43	0.34
Vanilla, resnet18	0.38	0.41	0.29	0.29	1.00	0.68	0.43	0.42
AT, vgg19	0.78	0.78	0.84	1.00	0.41	0.39	0.35	0.39
AT, vgg16	0.85	0.82	1.00	0.82	0.39	0.41	0.36	0.35
AT, resnet34	0.94	1.00	0.93	0.92	0.56	0.58	0.55	0.56
AT, resnet18	1.00	0.94	0.93	0.92	0.51	0.52	0.52	0.56

Figure 4. **[Different Architectures, Penultimate Layer Shared Invariances; CIFAR10]** We find that in general ResNets have high shared invariances when trained using the same loss, but this shared invariance drops across training types. We also see that with AT, even models with different architectures converge to high shared invariances.

To test this, we train a ResNet18 on CIFAR10 using 3 distinct training losses, all of which have been shown to be highly robust to ℓ_p ball perturbations: Adversarial Training (AT) (Madry et al., 2019), TRADES (Zhang et al., 2019) and MART (Wang et al., 2019). TRADES and MART contains an additional hyperparameter β that is used to trade off between accuracy and adversarial robustness. We use $\beta = 0.1$ for our experiments, and additional details can be found in Appendix B. All 3 of these ResNets achieve similar clean and robust accuracy.

Surprisingly, we find that models trained using these methods only have mild levels of shared invariance, as indicated by lines of best fit around 0.5 (see Fig. 5). This is in contrast to the case of two ResNet18s trained using AT (see Fig. 2b) which leads to very high shared invariance. This shows that the differences shown in Fig. 5 are not due to stochasticity in training but rather due to the training methods themselves. Thus, while all of these methods achieve the same goal of robustness in an ℓ_p ball, they achieve so in very different ways. We see similar trends for other architectures too (results in Appendix D).

In summary, we find that shared invariance between models tends to decrease with increasing layer-depth and adversarial training significantly increases the degree of shared invariance. Models with architectures using residual connections exhibit a higher degree of shared invariance, whereas different methods of adversarial training do not necessarily lead to models with the same shared invariances.

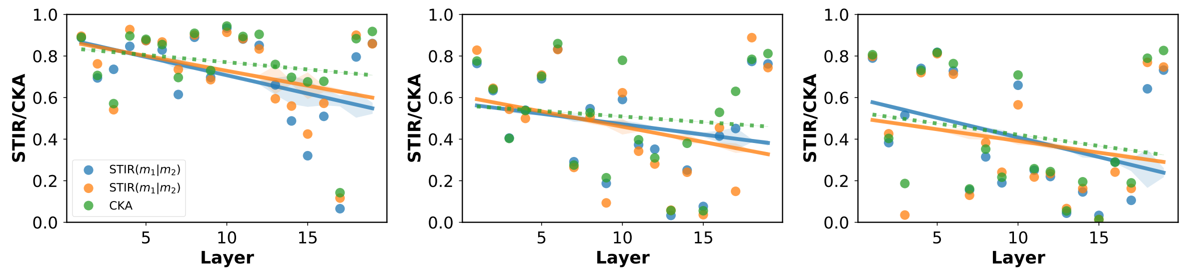
6. Related Work

6.1. Comparing Representations

A number of papers have proposed methods for measuring the similarity of representations in deep neural networks (Laakso & Cottrell, 2000; Li et al., 2016; Wang et al., 2018; Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019a). Our invariance measure leverages CKA (Kornblith et al., 2019a), however, we discuss in Sections 3.2 and 3.3 why the existing metrics themselves are unsuitable for measuring shared invariances between models. Recent work has identified lack of consistency between existing similarity measures and proposed a measure that is consistent (Ding et al., 2021). However, their proposed measure also measures similarity and does not take into account the invariances. It can thus be used in conjunction with our proposed measure, but not replace it for measuring invariance. There has been work on measuring shared invariance between NN representations and (black box) humans (Feather et al., 2019; Nanda et al., 2021), however, we consider a different problem of shared invariances between two NNs.

6.2. Understanding Representations

A lot of work on scrutinizing representations has been geared towards improving the interpretability of neural net-



(a) AT (m_1) TRADES (m_2) (b) AT (m_1) MART (m_2) (c) TRADES (m_1) vs MART (m_2)

Figure 5. **[Different Types of Adversarially Robust Training; ResNet18 on CIFAR10]** Comparing same model trained using 3 different adversarial training variants: AT (Madry et al., 2019), TRADES (Zhang et al., 2019) and MART (Wang et al., 2019). While these methods are geared towards the same goal of achieving invariance to ℓ_p perturbations, we find that other than TRADES and AT, these methods in fact do not have very high shared invariance (as opposed to similarity between two ResNet18s trained using AT, which have much higher values of shared invariance as shown in Fig. 2b). This shows that these methods achieve resistance to ℓ_p ball attacks in very different ways.

works (Mahendran & Vedaldi, 2014; Olah et al., 2017; Kim et al., 2018; Dosovitskiy & Brox, 2016a;b). We're particularly inspired by representation inversion (Mahendran & Vedaldi, 2014; Dosovitskiy & Brox, 2016b) which is a key component of our proposed measure. However, while the goals of all of these works is to be able to make a neural network more interpretable to humans, *i.e.,* to enable qualitative judgements about the network's behavior, our goal is to instead measure the degree of shared invariance between any two models, *i.e.,* to make quantitative statements. Recent work has used model stitching to compare representations (Bansal et al., 2021), but it is focused on better understanding of learned representations, rather than measuring robustness. Higgins et al. (2018) define disentanglement in representation learning by leveraging the concept of invariance. Their work, however, only provides a definition of disentanglement and no measure for the degree of invariance.

6.3. Robustness

Our work takes inspiration from many works on adversarial (Szegedy et al., 2013; Madry et al., 2019; Zhang et al., 2018; Ilyas et al., 2019), natural (Hendrycks & Dietterich, 2019; Geirhos et al., 2018) and distributional (Taori et al., 2020; Recht et al., 2019; Koh et al., 2021) robustness. However, in all these works, the reference model is (implicitly) assumed to be a human and the goal is to make a neural network follow the invariances of a human. In our work we explicitly characterize the reference model, which can be another neural network, that allows us to unify all the different notions of robustness.

7. Conclusion

We proposed a directional measure of shared invariance between representations that takes into account the invariances of the model that generated these representations. We showed how our measure faithfully estimates shared invariances where existing representation similarity methods may fail. Furthermore, we showed how our measure can be used to derive interesting insights about deep learning. It will be interesting to explore this direction further in future work, revisiting earlier analysis based on previous representation similarity approaches (Nguyen et al., 2020; Raghu et al., 2021). Another interesting avenue to explore is how our measure can be used during training to encourage one model to follow similar invariances to another. This could be helpful to update a neural network in safety-critical applications by ensuring that the new network maintains the invariances of the original model.

Acknowledgements

AW acknowledges support from a Turing AI Fellowship under grant EP/V025279/1, The Alan Turing Institute, and the Leverhulme Trust via CFI. VN, TS, and KPG were supported in part by an ERC Advanced Grant "Foundations for Fair Social Computing" (no. 789373). VN and JPD were supported in part by NSF CAREER Award IIS-1846237, NSF D-ISN Award #2039862, NSF Award CCF-1852352, NIH R01 Award NLM013039-01, NIST MSE Award #20126334, DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, ARPA-E Award #4334192 and a Google Faculty Research Award. Finally, we would like to thank anonymous reviewers for their constructive feedback and suggestions for experiments in Section 4.

References

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In Blockeel, H., Kersting, K., Nijssen, S., and Železný, F. (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 387–402. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40994-3.

Ding, F., Denain, J.-S., and Steinhardt, J. Grounding representation similarity with statistical testing. *arXiv preprint arXiv:2108.01661*, 2021.

Dosovitskiy, A. and Brox, T. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29:658–666, 2016a.

Dosovitskiy, A. and Brox, T. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4829–4837, 2016b.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pp. 1802–1811. PMLR, 2019.

Fawzi, A. and Frossard, P. Manitest: Are classifiers really invariant? *CoRR*, abs/1507.06535, 2015. URL http://arxiv.org/abs/1507.06535.

Feather, J., Durango, A., Gonzalez, R., and McDermott, J. Metamers of neural networks reveal divergence from human perceptual systems. *Advances in Neural Information Processing Systems*, 32, 2019.

Geirhos, R., Temme, C. R. M., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/0937fb5864ed06ffb59ae5f9b5ed67a9-Paper.pdf.

Golan, T., Raju, P. C., and Kriegeskorte, N. Controversial stimuli: Pitting neural networks against each other as models of human cognition. *Proceedings of the National Academy of Sciences*, 117(47):29330–29337, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1912334117. URL https://www.pnas.org/content/117/47/29330.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples, 2015.

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

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Higgins, I., Amos, D., Pfau, D., Racanière, S., Matthey, L., Rezende, D. J., and Lerchner, A. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230, 2018. URL http://arxiv.org/abs/1812.02230.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, 2019. URL https://papers.nips.cc/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019a.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2661–2671, 2019b.

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

Laakso, A. and Cottrell, G. Content and cluster analysis: assessing representational similarity in neural systems. *Philosophical psychology*, 13(1):47–76, 2000.

Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. Convergent learning: Do different neural networks learn the same representations?, 2016.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks, 2019.

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them, 2014.

Merriam-Webster. Robust. In *Merriam-Webster.com dictionary*. 2022. URL https://www.merriam-webster.com/dictionary/robustness.

Morcos, A. S., Raghu, M., and Bengio, S. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 5732–5741, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/a7a3d70c6d17a73140918996d03c014f-Abstract.html.

Nanda, V., Majumdar, A., Kolling, C., Dickerson, J. P., Gummadi, K. P., Love, B. C., and Weller, A. Exploring alignment of representations with human perception. *CoRR*, abs/2111.14726, 2021. URL https://arxiv.org/abs/2111.14726.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? *arXiv preprint arXiv:2008.11687*, 2020.

Nguyen, T., Raghu, M., and Kornblith, S. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*, 2020.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Pearl, J. *Causality*. Cambridge University Press, 2 edition, 2009. ISBN 978-0-521-89560-6.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6078–6087, 2017. ISBN 9781510860964.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/d8330f857a17c53d217014ee776bfd50-Paper.pdf.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Wang, L., Hu, L., Gu, J., Wu, Y., Hu, Z., He, K., and Hopcroft, J. Towards understanding learning representations: To what extent do different neural networks learn the same representation. *arXiv preprint arXiv:1810.11750*, 2018.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482. PMLR, 2019.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

A. Representation Similarity Measures

A recent line of work has studied measures for the similarity of representations in deep neural networks. Given two models m_1 and m_2 and a set Z of input points to both, all of these measures quantify the degree of representational similarity between two sets of representations $X = m_1(Z) \in \mathbb{R}^{n\times d_1}$ and $Y = m_2(Z) \in \mathbb{R}^{n\times d_2}$ as $\mathrm{S_r}(X, Y)$ and are defined as follows:

SVCCA (Raghu et al., 2017) is computed via the following steps:

1. Compute X', Y' by pruning X, Y using SVD to only retain the first k_1 and k_2 principal components that are necessary to retain 99% of the variance, respectively.
2. Perform Canonical Correlation Analysis CCA(X', Y') yielding $k = \min(k_1, k_2)$ correlation coefficients $\rho_1, \ldots, \rho_k$.
3. Return $\mathrm{S_r}(X, Y) = \frac{1}{k}\sum_{i=1}^{k}\rho_i$

PWCCA (Morcos et al., 2018) is computed via the following steps:

1. Perform Canonical Correlation Analysis CCA(X, Y) yielding $k = min(d_1, d_2)$ correlation coefficients $\rho_1, \ldots, \rho_k$ and CCA vectors $h_1, \ldots, h_k$
2. Compute weights $\alpha_i = \sum_{j=1}^{d_1} |\langle h_i, z_j\rangle|$, where the z_j's are the d_1 columns of X
3. Return $\mathrm{S_r}(X, Y) = \frac{\sum_{i=1}^{k}\alpha_i\rho_i}{\sum_{i=1}^{k}\alpha_i}$

(Linear) CKA (Kornblith et al., 2019a) is computed via the following steps:

1. Compute similarity matrices $K = XX^T$, $L = YY^T$
2. Compute normalized versions $K' = HKH$, $L' = HLH$ of the similarity matrix using centering matrix $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$
3. Return CKA$(X, Y) = \frac{\text{HSIC}(K,L)}{\sqrt{\text{HSIC}(K,K)\text{HSIC}(L,L)}}$, where HSIC$(K, L) = \frac{1}{(n-1)^2}$flatten$(K')\cdot$ flatten(L')

B. Model Architecture, Hardware, Training, and Other Details

We use ResNet18, ResNet34 (He et al., 2016), VGG16 and VGG19 (Simonyan & Zisserman, 2014) trained on CIFAR10/100 (Krizhevsky et al., 2009) using the standard crossentropy loss and other adversarially robust training methods such as AT, TRADES and MART. All of our experiments are performed on standard models and datasets that can fit on standard GPUs. We also attach our code to reproduce the numbers. For all purposes of adversarial training we use the ℓ_2 threat model with $\epsilon = 1.0$ (see (Madry et al., 2019) for details). Additionally TRADES and MART require another hyperparameter β (that balances adversarial robustness and clean accuracy) which we set to 1.0 for our experiments.

C. STIR Faithfully Measures Shared Invariance

To confirm that STIR is correct in assigning different scores to two (same) models trained from different initializations, we generate controversial stimuli (Golan et al., 2020) for all configurations in Table 3: $(\mathrm{m^{(Vanilla)}}_1, \mathrm{m^{(Vanilla)}}_2)$, $(\mathrm{m^{(AT)}}_1, \mathrm{m^{(AT)}}_2)$, and $(\mathrm{m^{(Vanilla)}}_1, \mathrm{m^{(AT)}}_2)$. Such stimuli are generated by solving the following optimization for a training data point x and any two models $m_1$and m_2:

$$\text{argmin}_{x_c}\ ||m_1(x) - m_1(x_c)||_2 - ||m_2(x) - m_2(x_c)||_2 \quad (8)$$

Here we assume $m_1(.)$ and $m_2(.)$ are penultimate layer representations of the respective models. This process generates x_c for every point x such that $||m_1(x) - m_1(x_c)||_2$ is low and $||m_2(x) - m_2(x_c)||_2$ is high. Since these x_c are "perceived" very differently by the two models (wrt the original x), they are called *controversial stimuli*. We use the empirical mean of the amount of perturbation needed (*i.e.*,$\delta = \mathbb{E}_{x_c,x}[||x - x_c||_2]$) as a measure of ease of generating controversial stimuli [3]. For a pair of models (m_1, m_2) where it's easy to generate such x_c, the shared invariance should be low. We see that this is indeed the case as indicated by numbers in Table 3.

D. Experiments: Insights using STIR

D.1. Role of Different Random Initialization, Different Training Datasets

Fig 6&7 shows results for different random initializations and different datasets respectively. We see similar trends hold for deeper models such as ResNet34 and VGG19.

D.2. Different Architectures, Penultimate Layer

Fig 8 shows that trends for STIR hold across different choice of seeds. We also see that in contrast CKA assigns high value across the board and is thus not a faithful measure of shared invariance.

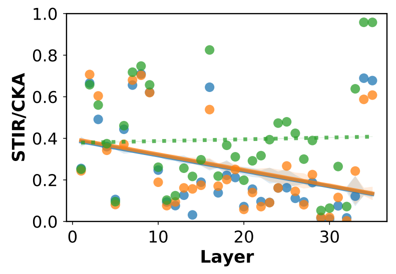
D.3. Different Adversarial Training Methods

Fig 9 shows results on VGG16. These are much higher than values for ResNet18, thus showing that these methods achieve robustness differently across architectures. For

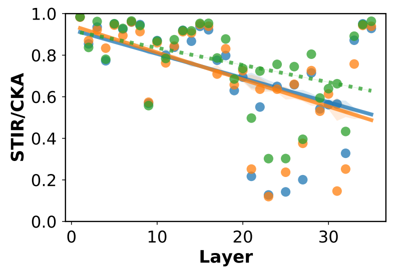
[3]High(er) values of δ indicate it was hard(er) to generate controversial stimuli

	ResNet18 Vanilla (m_1), ResNet18 Vanilla (m_2)		ResNet18 AT (m_1), ResNet18 AT (m_2)		ResNet18 AT (m_1), ResNet18 Vanilla (m_2)	
	$m_1 \| m_2$	$m_2 \| m_1$	$m_1 \| m_2$	$m_2 \| m_1$	$m_1 \| m_2$	$m_2 \| m_1$
STIR	$0.605_{\pm 0.013}$	$0.562_{\pm 0.023}$	$0.934_{\pm 0.003}$	$0.939_{\pm 0.002}$	$0.405_{\pm 0.020}$	$0.509_{\pm 0.011}$
STIR$_{adv}$	$0.085_{\pm 0.004}$	$0.064_{\pm 0.004}$	$0.096_{\pm 0.007}$	$0.078_{\pm 0.005}$	$0.054_{\pm 0.004}$	$0.070_{\pm 0.004}$
CKA	$0.967_{\pm 0.000}$		$0.937_{\pm 0.000}$		$0.536_{\pm 0.000}$	
Acc($m_1(X'), m_2(X')$)	$0.521_{\pm 0.061}$	$0.347_{\pm 0.029}$	$0.891_{\pm 0.007}$	$0.901_{\pm 0.002}$	$0.140_{\pm 0.028}$	$0.555_{\pm 0.012}$
Δ ($= \frac{1}{n}\sum \|\|X - X'\|\|_2$)	$8.588_{\pm 0.491}$	$8.570_{\pm 0.474}$	$17.106_{\pm 3.204}$	$16.210_{\pm 2.193}$	$19.270_{\pm 2.202}$	$7.368_{\pm 0.602}$

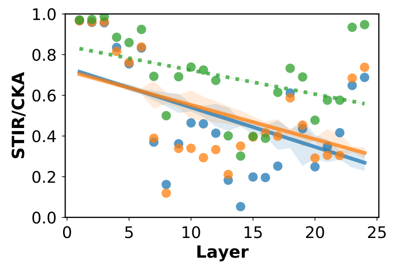
Table 3. **[STIR faithfully estimates shared invariance]** Here the two ResNet18s in each column are trained on CIFAR10 with different random initializations, holding every other hyperparameter constant. 1.) For two such models trained using the vanilla crossentropy loss (left), interestingly, we find that STIR highlights a lack of shared invariance, whereas CKA overestimates this value; 2.) when both models are trained using adversarial training (Madry et al., 2019) (middle) STIR faithfully estimates high shared invariance; 3.) Finally STIR is able show how having a directional measure can bring out the differences when comparing a model trained with vanilla loss and adv training (right), whereas CKA being unidirectional cannot derive these insights. All numbers are computed over 1000 random samples from CIFAR10 training set and averaged over 5 runs. Additionally we show Δ between original inputs (X) and controversial stimuli (X') for a given configuration of m_1 and m_2. We see that controversial stimuli are "hard" to generate for higher STIR scores. Here hardness is indicated by the amount of ℓ_2 perturbation needed to generate controversial stimuli.



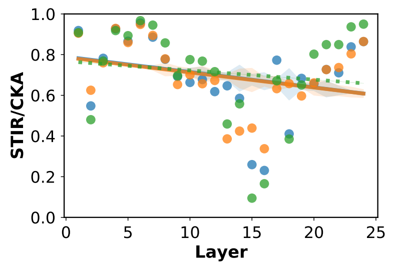
(a) 2 ResNet34 with different init, Vanilla



(b) 2 ResNet34 with different init, AT



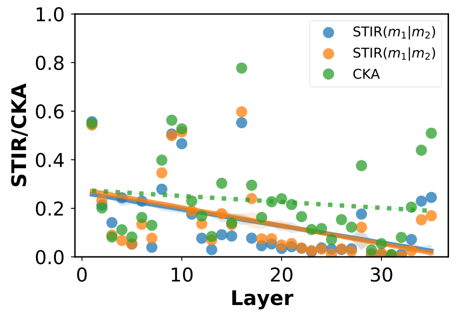
(c) 2 VGG19 with different init, Vanilla



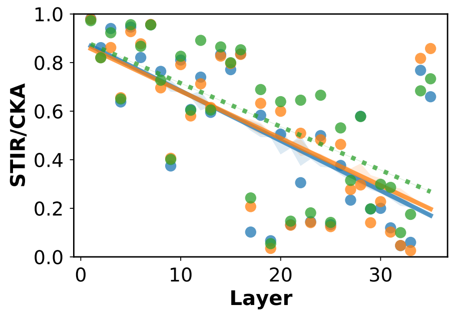
(d) 2 VGG19 with different init, AT

Figure 6. **[Role of Random Initialization; CIFAR10]** For models trained using the Vanilla crossentropy loss, we find that only initial layers have high shared invariances. However, when the training procedure explicitly introduces invariances (*e.g.,*adversarial training), then all layers converge to having similarly high shared invariances. We see similar trends for ResNet34 and VGG19 here.

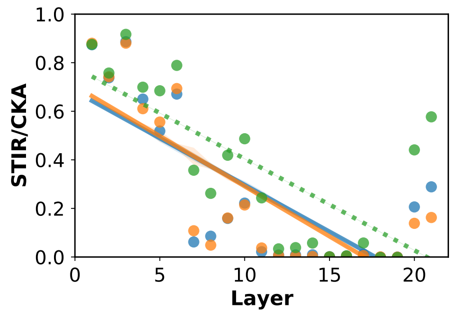
AT and MART, we see lower shared invariances even for VGG16.s



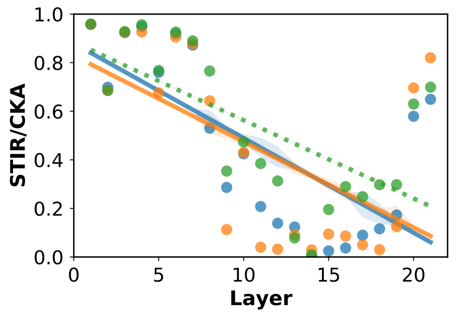
(a) m_1= ResNet34 on CIFAR10 w ERM
m_2= ResNet34 on CIFAR100 w ERM



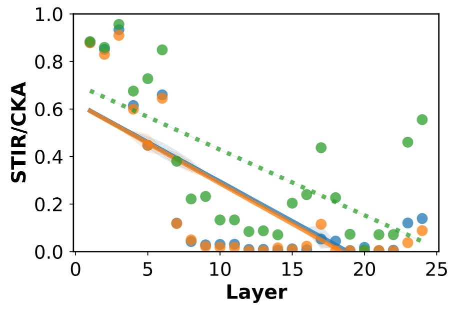
(b) m_1= ResNet34 on CIFAR10 w AT
m_2= ResNet34 on CIFAR100 w AT



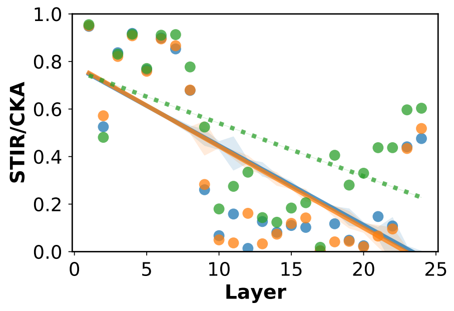
(c) m_1= VGG16 on CIFAR10 w ERM
m_2= VGG16 on CIFAR100 w ERM



(d) m_1= VGG16 on CIFAR10 w AT
m_2= VGG16 on CIFAR100 w AT

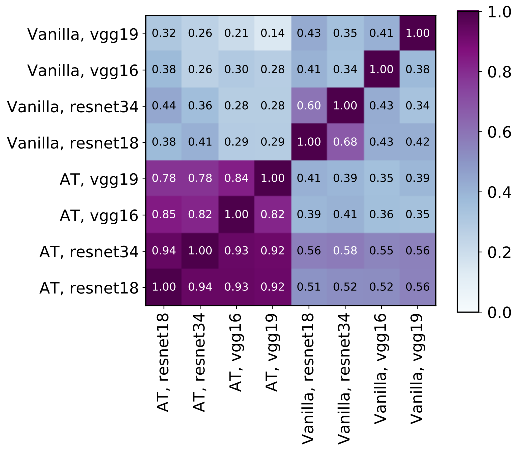


(e) m_1= VGG19 on CIFAR10 w ERM
m_2= VGG19 on CIFAR100 w ERM

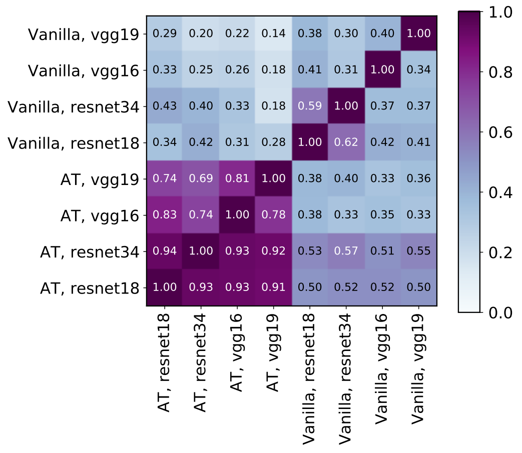


(f) m_1= VGG19 on CIFAR10 w AT
m_2= VGG19 on CIFAR100 w AT

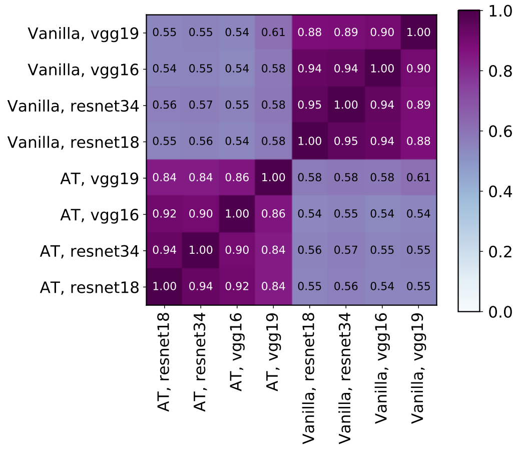
Figure 7. **[Different Datasets; ResNet34, VGG16 and VGG19 on CIFAR10 and CIFAR100]** Similar to the finding of (Kornblith et al., 2019a) we find that across datasets, initial layers tend to have more shared invariances. However, we also find that shared invariances increases substantially for all layers with adversarial training (AT).



(a) STIR- completely random seeds

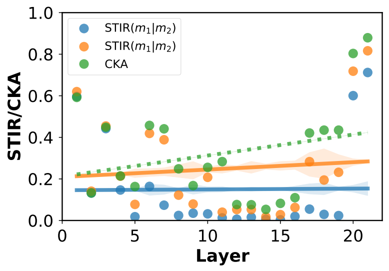


(b) STIR- seeds sampled from $\mathcal{N}(0, 1)$

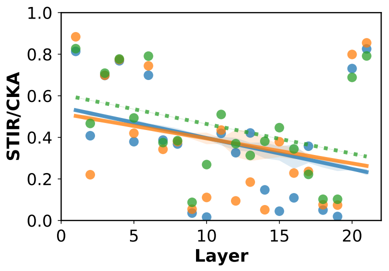


(c) CKA

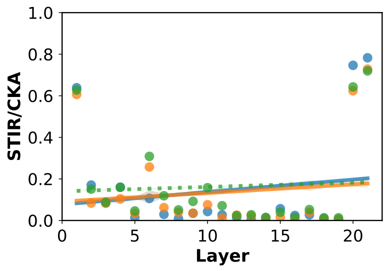
Figure 8. **[Different Architectures, Penultimate Layer Shared Invariances; CIFAR10]** We find that in general ResNets have high shared invariances when trained using the same loss, but this shared invariance drops across training types. We also see that with AT, even models with different architectures converge to high shared invariances. Here additionally we show comparison with CKA which does not provide any faithful estimate of shared robustness and is unformly high for similarly trained models. We also see that this trend holds even when different seeds are used to generate X'.



(a) AT (m_1) TRADES (m_2)



(b) AT (m_1) MART (m_2)



(c) TRADES (m_1) vs MART (m_2)

Figure 9. **[Different Types of Adversarially Robust Training; VGG16 on CIFAR10]** We se much high levels of shared invariance for VGG16 than for ResNet18. Even for VGG16, AT and MART achieve ell_p ball robustness in different ways.