WHAT KNOWLEDGE GETS DISTILLED IN KNOWLEDGE DISTILLATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation aims to transfer useful information from a teacher network to a student network, with the primary goal of improving the student's performance for the task at hand. Over the years, there has a been a deluge of novel techniques and use cases of knowledge distillation. Yet, despite the various improvements, there seems to be a glaring gap in the community's fundamental understanding of the process. Specifically, what is the knowledge that gets distilled in knowledge distillation? In other words, in what ways does the student become similar to the teacher? Does it start to localize objects in the same way? Does it get fooled by the same adversarial samples? Does its data invariance properties become similar? Our work presents a comprehensive study to try to answer these questions and more. Our results, using image classification as a case study and three state-of-the-art knowledge distillation techniques, show that knowledge distillation methods can indeed indirectly distill other kinds of properties beyond improving task performance. And while we believe that understanding the distillation process is important in itself, we also demonstrate that our results can pave the path for important practical applications as well.

1 INTRODUCTION

Knowledge distillation, first introduced in Bucila et al. (2006); Hinton et al. (2014b), is a procedure of training neural networks in which the 'knowledge' of a teacher is transferred to a student. The thesis is that such a transfer (i) is possible and (ii) can help the student learn additional useful representations. The seminal work by Hinton et al. (2014b) demonstrated its effectiveness by making the student imitate the teacher's class probability outputs for an image. This ushered in an era of knowledge distillation algorithms, including those that try to mimic intermediate features of the teacher (Romero et al., 2015; Zagoruyko & Komodakis, 2017; Huang & Wang, 2017), or preserve the relationship between samples as modeled by the teacher (Park et al., 2019; Yim et al., 2017), among others.

While thousands of papers have been published on different techniques and ways of using knowledge distillation, there appears to be a glaring gap in the community's fundamental understanding of it. Yes, it is well-known that the student's performance on the task at hand can be improved with the help of a teacher. But what exactly is the so-called *knowledge* that gets distilled during the knowledge distillation process? For example, does distillation make the student look at similar regions as the teacher when classifying images? If one crafts an adversarial image to fool the teacher, is the student also more prone to getting fooled by it? If the teacher is invariant to a certain change in data, is that invariance also transferred to the student? Does the teacher's behavior on out-of-distribution data get transferred to the student? If the teacher is shape-biased, does this property get transferred to the student, who otherwise would have been texture-biased? We argue that such questions have not been answered in the existing literature.

The need for such answers has become particularly relevant because there have been studies which present some surprising findings about the distillation process. Cho & Hariharan (2019) showed that performing knowledge distillation with a bigger teacher does not necessarily improve the student's performance over that with a smaller teacher, and thus raised questions about the effectiveness of the distillation procedure in such cases. Stanton et al. (2021) showed that the agreement between the teacher and distilled student's predictions on test images is not that different to the agreement of those between the teacher and an independently trained student, raising further doubts about how knowledge distillation works, if it works at all.

Our goal is hence to shed some light on the 'dark knowledge' (Hinton et al., 2014a) that gets distilled in knowledge distillation. In this work, we present a comprehensive study tackling the above questions. We use the task of image classification as a case study, and analyze three popular state-of-the-art knowledge distillation methods (Hinton et al., 2014b; Romero et al., 2015; Tian et al., 2020). Many of our findings are quite surprising. For example, by simply mimicking the teacher's output on ImageNet images using the method of Hinton et al. (2014b), the student can inherit many implicit properties of the teacher. It can gain the adversarial vulnerability that the teacher has. If the teacher is invariant to color, the student also improves its invariance to color, even without explicitly being trained to be that way. While the specific transferred properties and their extent differ across different distillation methods and teacher-student architectures, our study reveals that there is much more happening in the background of the distillation process than what meets the eye: distillation makes the student extract similar intermediate representations as the teacher, something we study quantitatively. We show that this has practical implications, a good one and a bad one. The good is that a student can inherit many useful properties of a teacher even without being exposed to it explicitly, which saves computation. The bad is that an otherwise *fair* student can inherit biases from an unfair teacher. We believe that the existence of these two contrasting sides of the distillation process should motivate understanding this topic better.

2 Related work

Model compression (Bucila et al., 2006) first introduced the idea of knowledge distillation by compressing an ensemble of models into a smaller network. Hinton et al. (2014b) took the concept forward for modern deep learning by training the student to mimic the teacher's output probabilities. Some works train the student to be similar to the teacher in the intermediate feature spaces (Romero et al., 2015; Zagoruyko & Komodakis, 2017). Others train the student to mimic the relationship between samples produced by the teacher (Park et al., 2019; Tung & Mori, 2019; Peng et al., 2019), so that if two samples are close/far in the teacher's representation, they remain close/far in the student's representation. Contrastive learning has recently been shown to be an effective distillation objective in Tian et al. (2020). More recently, Beyer et al. (2021) present practical tips for performing knowledge distillation; e.g., providing the same view of the input to both the teacher and the student, and training the student long enough through distillation. Finally, the benefits of knowledge distillation have been observed even if the teacher and the student have the same architecture (Furlanello et al., 2018; Zhang et al., 2018). For a more thorough survey of knowledge distillation, see Gou et al. (2021). In this work, we choose to study three state-of-the-art methods, each representative of the output-based, feature-based, and contrastive-based families of distillation approaches.

Although no comprehensive study exists in trying to understand knowledge distillation, there have been a few papers that present some surprising results. Cho & Hariharan (2019) challenge the assumption that a teacher with better test accuracy, which is typically a bigger network, would have more knowledge to pass on to the student. Instead, it shows that distillation with a bigger teacher does not necessarily increase a student's accuracy compared to having a relatively smaller teacher. More recent work shows that the agreement between the predictions of a teacher and student is not necessarily much higher than that between the teacher and an independent student (Stanton et al., 2021). However, even if the teacher and student do not become similar along one axis, they could still become more similar along different ones. Our study thus investigates the various other ways in which knowledge can get distilled into a student.

3 Distillation methods studied

To ensure that our findings are general and cover a range of distillation techniques, we select standard methods representative of three families of distillation techniques: output-based (Hinton et al., 2014b), feature-based (Romero et al., 2015), and contrastive-based (Tian et al., 2020). The objectives of these methods, described below, are combined with the cross entropy loss $\mathcal{L}_{\text{CLS}}(\mathbf{z_s}, \mathbf{y}) := -\sum_{j=1}^{c} y_j \log \sigma_j(\mathbf{z_s})$, where $\mathbf{y}$ is the ground-truth one-hot label vector, $\mathbf{z_s}$ is the student's logit output, $\sigma_j(\mathbf{z}) = \exp(z_j)/\sum_i \exp(z_i)$ is the softmax function, and c is the number of classes.

(1) **KL:** Hinton et al. (2014b) proposed to use the soft labels produced by the teacher as an additional target for the student to match, apart from the (hard) ground-truth labels. This is done by minimizing the KL-divergence between the predictive distributions of the student and the teacher:

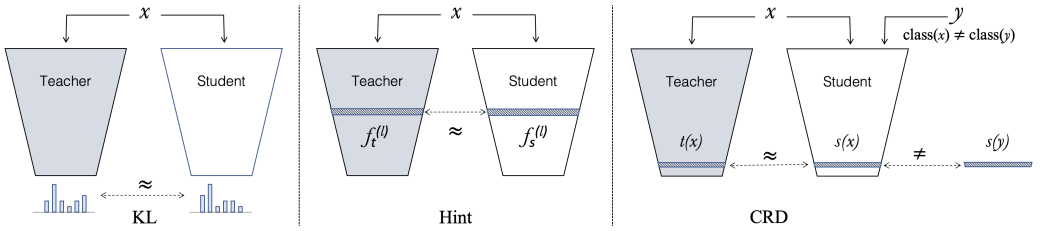


Figure 1: Distillation methods used in this work. (i) KL (Hinton et al., 2014b): student tries to mimic class probabilities. (ii) $Hint$ (Romero et al., 2015): student tries to mimic features at an intermediate layer. (iii) CRD (Tian et al., 2020): features obtained by the student and teacher for the same image constitute a positive pair, and those obtained for different classes make up a negative pair.

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{z_s}, \mathbf{z_t}) := -\tau^2 \sum_{j=1}^{c} \sigma_j \left(\frac{\mathbf{z_t}}{\tau}\right) \log \sigma_j \left(\frac{\mathbf{z_s}}{\tau}\right), \tag{1}$$

where $\mathbf{z_t}$ is the logit output of the teacher, and τ is a scaling temperature. The overall loss function is $\gamma\mathcal{L}_{\mathrm{CLS}} + \alpha\mathcal{L}_{\mathrm{KL}}$, where γ and α are balancing parameters. We refer to this method as KL.

(2) **Hint:** FitNets (Romero et al., 2015) makes the student's intermediate features ($\mathbf{f_s}$) mimic those of the teacher's ($\mathbf{f_t}$) for an image x, at some layer l. It first maps the student's features (with additional parameters r) to match the dimensions of the teacher's features, and then minimizes their mean-squared error:

$$\mathcal{L}_{Hint}(\mathbf{f_s^{(l)}}, \mathbf{f_t^{(l)}}) = \frac{1}{2}||\mathbf{f_t^{(l)}} - r(\mathbf{f_s^{(l)}})||^2 \tag{2}$$

The overall loss is $\gamma\mathcal{L}_{\mathrm{CLS}} + \beta\mathcal{L}_{\mathrm{Hint}}$, where γ and β are balancing parameters. Romero et al. (2015) termed the teacher's intermediate representation as $Hint$, and we adopt this name.

(3) **CRD:** Contrastive representation distillation (Tian et al., 2020) proposed the following. Let $s(x)$ and $t(x)$ be the student's and teacher's penultimate feature representation for an image x. If x and y are two images from different categories, then $s(x)$ and $t(x)$ should be similar (positive pair), and $s(x)$ and $t(y)$ should be dissimilar (negative pair). A key for better performance is the number of negative samples N used for each image. Tian et al. (2020) use a memory bank of all the training data, which is constantly updated, to draw a large number of negative samples on the fly:

$$\mathcal{L}_{\mathrm{CRD}} = -\log h(s(x), t(x)) - \sum_{j=1}^{N} \log(1 - h(s(x), t(y_j))) \tag{3}$$

where $h(a, b) = (e^{a \cdot b/\tau})/(e^{a \cdot b/\tau} + \frac{N}{M})$, M is the size of the entire training data, τ is a scaling temperature, and $\cdot$ is the dot product. We use CRD to refer to this method. All other implementation details (e.g., temperature for *KL*, layer index for *Hint*) can be found in appendix (Sec. A).

4 Experiments

We now discuss our experimental design. We use the image classification task as our case study, and use ImageNet (Deng et al., 2009), MNIST (LeCun, 1998), as well as images from artistic domains (Geirhos et al., 2021). To reach conclusions that are generalizable across different architectures and datasets, and robust across independent runs, we experiment with a variety of teacher-student architectures, and tune the hyperparameters so that the distillation objective improves the test performance of the student compared to independent training. For each setting, we average the results over two independent runs; each time, the distilled student and independent student (without distillation) are trained using the same random seed. We report the top-1 accuracy of all the models—teacher, independent, and distilled students—in the appendix (Sec. D). The notation $\texttt{Net}_1 \rightarrow \texttt{Net}_2$ indicates distilling the knowledge of $\texttt{Net}_1$ (teacher) into $\texttt{Net}_2$ (student).

4.1 Does localization knowledge get distilled?

We start by studying whether the localization properties of a teacher transfers to a student through knowledge distillation. For example, suppose that the teacher classifies an image as a cat by focusing

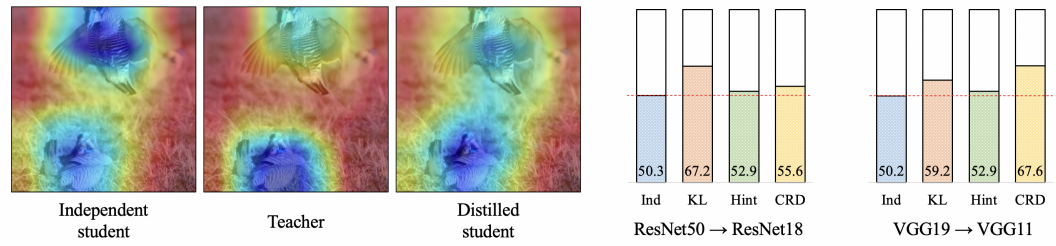


Figure 2: **Left:** An example of how the distilled student can focus on similar regions as the teacher while classifying an image. **Right:** % where teacher's CAM is more similar to the distilled student's CAM than to the independent student's CAM. The red line indicates chance performance (50%).

on its face, and an independent student classifies it as a cat by focusing on its fur. After knowledge distillation, will the student focus *more* on the face when classifying the image as a cat?

Experimental setup: We train three models for ImageNet classification: a teacher, a distilled student, and an independent student (without distillation). For each network, we obtain the class activation map (CAM) of the ground-truth class for each of (randomly sampled) 5000 test images, using Grad-CAM (Selvaraju et al., 2019), which visualizes the pixels that a classifier focuses on for a given class. We then compute how often (in terms of % of test images) the teacher's CAM is more similar (using cosine similarity) to the distilled student's CAM than to the independent student's CAM. A method that leads to $>$50% means that, on average, the distilled student's CAMs become more similar to those of the teacher than without distillation. As a sanity check, we also compute the same metric between the teacher's CAM and the CAMs of two independent students (without distillation) trained with two random seeds (*Ind*). These two models should be equally distant from the teacher, hence achieving a score of $\sim$50%.

Results: Fig. 2 (right) shows the results for two configurations of teacher-student architectures: (i) ResNet50 $\rightarrow$ ResNet18 and (ii) VGG19 $\rightarrow$ VGG11. For *KL* and *CRD*, we observe a significant increase in similarity compared to random chance (as achieved by the *Ind* baseline). *Hint* also shows a consistent increase, although the magnitude is not as large.

Discussion: This experiment shows that having access to the teacher's class-probabilities, i.e. confidence for the ground-truth class, can give information on where the teacher is focusing on while making the classification decision. This corroborates, to some degree, the result obtained in the Grad-CAM paper (Selvaraju et al., 2019), which showed that if the network is very confident of the presence of an object in an image, it focuses on a particular region (Fig. 1(c) in Selvaraju et al. (2019)), and when it is much less confident, it focuses on some other region (Fig. 7(d) in Selvaraju et al. (2019)). Fig. 2 (left) shows a sample test image and the corresponding CAMs produced by the independent student (left), teacher (middle), and distilled student (right) for the ground-truth class. The distilled student looks at similar regions as the teacher, and moves away from the regions it was looking at initially (independent student). Importantly, we are not interested in the correctness of any network's CAM; rather, we are using CAM as a tool to study whether a distilled student's CAMs become similar to those of the teacher's. And our analysis indicates this to be the case for all three distillation techniques, albeit with varying degrees.

4.2 Does adversarial vulnerability get distilled?

Next, we study the transfer of a different kind of property: the vulnerability to adversarial images. Specifically, if we design an adversarial image to fool the teacher, then will that same image fool the distilled student more than the independent student?

Experimental setup: We train three models for ImageNet classification: a teacher, a distilled student, and an independent student (without distillation). Given 5000 random test images, we convert each image I into an adversarial image I^{adv} using iterative FGSM (Goodfellow et al., 2014; Kurakin et al., 2016); see appendix (Sec. B) for details of this process. The goal of this conversion ($I \rightarrow I^{adv}$) is to fool the teacher, and is considered successful if the class predicted by the teacher for I^{adv} is different than what it predicts for I. Fooling rate is then defined as the fraction of adversarial images which succeed at this task. In our experiments, we use only this fraction of adversarial images, $\{I_1, I_1^{adv}\}$,

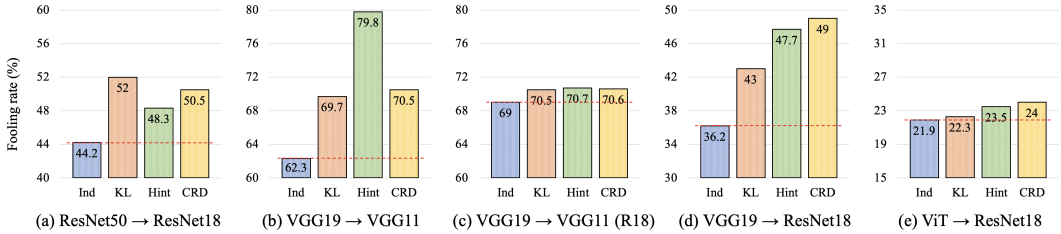


Figure 3: The images which fool the teacher fool the distilled student more than the independent student in **(a)**, **(b)** and **(d)**, but not in **(e)**. If the adversarial attack is generated using a foreign network (ResNet18, **(c)**), it fails to convincingly fool the distilled student more than the independent one.

$\{I_2, I_2^{adv}\}$ ..., which fool the teacher, and then ask if those images fool the distilled student more than the independent student.

Results: We evaluate four teacher-student configurations: (i) ResNet50 $\rightarrow$ ResNet18, (ii) VGG19 $\rightarrow$ VGG11, (iii) VGG19 $\rightarrow$ ResNet18 and (iv) ViT (ViT-b-32) (Dosovitskiy et al., 2020) $\rightarrow$ ResNet18. The adversarial attack is successful in $\sim$85% of the cases (among 5000 test images) for all the teachers. Fig. 3 shows the fooling rate (y-axis) when applying these successful adversarial images that fooled the teacher to different types of students (x-axis). We see that for ResNet50 $\rightarrow$ ResNet18, the ability to fool the independent student drops to 44.2%, which is expected since the adversarial images are not designed for the independent student. With the distilled student, we see a consistent increase in the fooling rate relative to the independent student, across all distillation methods (48%-52%). A similar trend holds for VGG19 $\rightarrow$ VGG11 and VGG19 $\rightarrow$ ResNet18; Fig. 3 (b, d). Finally, when distillation is done from a transformer to a CNN (ViT $\rightarrow$ ResNet18) the fooling rates remain similar for the independent and distilled students; Fig. 3 (e). In this setting, we do not see the student becoming similar to the teacher, to the extent observed for a CNN $\rightarrow$ CNN distillation.

Discussion: This result is surprising. Iterative FGSM is a white box attack, which means that it has full access to the target model (teacher), including its weights and gradients. Thus, the adversarial examples are specifically crafted to fool the teacher. In contrast, the student never has direct access to the weights and gradients of the teacher, regardless of the distillation method. Yet, by simply trying to mimic the teacher's soft probabilities (KL) or an intermediate feature layer ($Hint$, CRD), the student network inherits, to some extent, the particular way that a teacher is fooled.

We conduct an additional study to ensure that the reason the distilled student is getting fooled more is because it is being attacked specifically by its *teacher's* adversarial images; i.e., if those images are designed for some other network, would the difference in fooling rates still be high? We test this in the VGG19 $\rightarrow$ VGG11 setting. This time we generate adversarial images ($I \rightarrow I^{adv}$) to fool an ImageNet pre-trained ResNet18 network, instead of VGG19 (the teacher). We then use those images to attack the same VGG11 students from the VGG19 $\rightarrow$ VGG11 setting. In Fig. 3 (c), we see that the fooling rates for the independent and distilled students remain similar. This indicates that distillation itself does not make the student more vulnerable to *any* adversarial attack, and instead, an increase in fooling rate can be attributed to the student's inheritance of the teacher's adversarial vulnerability.

4.3 Does invariance to data transformations get distilled?

So far, we have studied whether the response properties on single images get transferred from the teacher to the student. Now, we analyze whether the response to *changes in images* gets transferred. Specifically, the teacher might be invariant to certain changes in data, either learned explicitly through data augmentation or implicitly due to architectural choices. In this section, we study whether such properties are transferred during distillation.

Experimental setup: We study color invariance as a transferable property here. (The appendix (C.4) contains a similar experiment where we study invariance to random shifts in images instead and find a similar result). We train three models for ImageNet classification: a teacher, a distilled student, and an independent student (without distillation). While training the teacher, we add color jitter in addition to the standard data augmentations (random crops, horizontal flips). Specifically, we alter an image's brightness, contrast, saturation, and hue, with magnitudes sampled uniformly in [0, 0.4] for the first three, and in [0, 0.2] for hue. This way, the teacher gets to see the same image

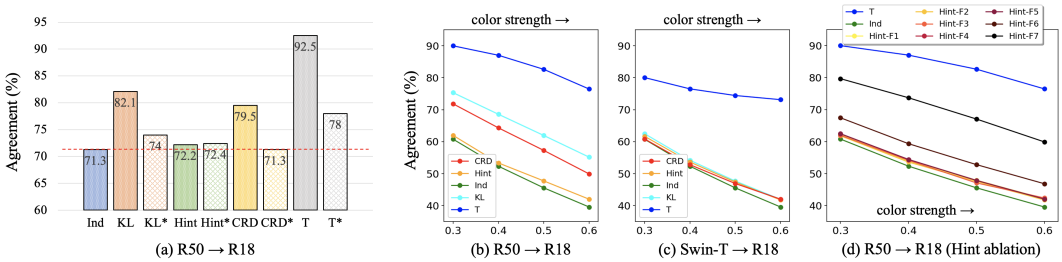


Figure 4: **(a)** Agreement between two images with different color properties by different models, where * indicates distillation done by a teacher T* not trained to be color invariant. **(b, c)** Agreement between two images having increasingly different color properties. **(d)** Studying the effect of different layers l used for $Hint$ distillation.

with different color properties across training iterations. When training the student, we only use the standard data augmentations *without* color jittering. That is, we want to study whether the distilled student, which is trained *without* color augmentation, inherits that property through the teacher that is trained *with* color augmentation. And if the student does become invariant to color changes, then it should produce the same classification label for two different color augmentations of the same image. In contrast, an independent student is less likely to produce the same classification label, since it never learned about color invariance (either directly or through the teacher).

Results: We start with the ResNet50 → ResNet18 configuration. After training, we evaluate the models on 50k ImageNet validation images. For each image X, we construct its augmented version X' by altering the color properties in the same way as done while training the teacher. Fig. 4 (a) depicts the agreement scores between X and X' (y-axis) for different models (x-axis). The teacher (T), being trained to have that property, achieves a high score of 92.5%. The independent student, which is not trained to be color invariant, has a much lower score of 71.3%. But, when the color-invariant teacher is used to perform distillation using KL or CRD, the agreement scores of the students jump up to 82.1% and 79.5% respectively. To ensure that this increase is not due to some regularizing property of the distillation methods that has nothing to do with the teacher, we repeat the experiment, this time with a ResNet50 teacher (T*) that is not trained with color augmentation. The new agreement scores for the distilled students (marked by *; e.g., KL*) drop considerably compared to the previous case. This is a strong indication that the student does inherit *teacher-specific* invariance properties during distillation.

In Fig. 4(b), we show that this trend in agreement scores (y-axis) holds even when the magnitude of change in brightness, contrast and saturation (x-axis) is increased to create X' from X. We repeat this experiment for Swin-tiny (a transformer) (Liu et al., 2021) → ResNet18 in Fig. 4(c). We again see some improvements in the agreement scores of the distilled students. This increase, however, is not as significant as that observed in the previous CNN → CNN setting. Throughout this work, we find that distilling the properties from a transformer teacher into a CNN student is difficult. The implicit biases introduced due to architectural differences between the teacher (transformer) and student (CNN) seem too big, as was studied in Raghu et al. (2021), to be overcome by current distillation methods.

We also note, however, that even $Hint$ has trouble distilling this knowledge with similar effectiveness as KL or CRD. Our initial guess for this is because the default $Hint$ asks the student to mimic the teacher at somewhere in the middle of the network. This is different than KL and CRD, which operate at deeper levels of the network. So, we perform multiple rounds of $Hint$ distillation as an ablation study. In each round, the student mimics a different feature of the teacher; starting from a coarse feature of resolution $56 \times 56 \times 64$ (F1) to the output scores (logits) of the teacher network (F7). We plot the agreement scores for this experiment in Fig. 4(d). In general, we observe that the choice of layer can have an impact on the amount of knowledge that gets transferred. In general, the agreement increases as we choose deeper features. However, why should the information about color invariance be better encoded in deeper rather than shallower layers is an open question.

Discussion: In sum, color invariance can be transferred during distillation to some extent by all the methods. This is quite surprising since the teacher is not used in a way which would expose any of its invariance properties to the student, or at least in any straightforward manner. Remember that all the student has to do during distillation is to match the teacher's features for an image X; i.e., the student

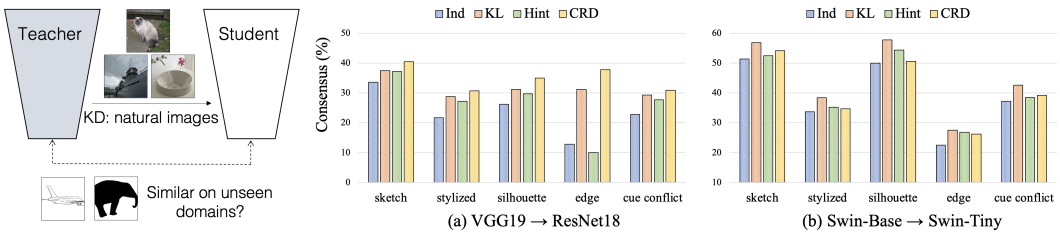


Figure 5: **Left:** Does knowledge transferred about one domain give knowledge about other unseen domains? **Right:** Consensus scores between teacher and student for images from unseen domains.

does not get to see its color augmented version, X', during training. If so, then why should it get to know how the teacher would have responded to X'? We try providing an answer in Sec. 5.

4.4 Does knowledge about unseen domains get distilled?

So far, our analysis has revolved around the original ImageNet dataset, something that was used to perform the distillation itself. So, does the teacher only transfer its knowledge pertaining to this domain, or also of domains it has never seen (Fig. 5 left)?

Experimental setup: To test this, we first perform distillation using two settings (i) VGG19 $\rightarrow$ ResNet18 and (ii) Swin-Base (Liu et al., 2021) $\rightarrow$ Swin-Tiny, where the training of the teacher, as well as distillation is done on ImageNet. During test time, we take an image from an unseen domain and see how frequently the student's and teacher's class predictions match for it (regardless of whether the predicted label is correct or incorrect), which we call the consensus score. For the unseen domains, we consider the five datasets proposed in Geirhos et al. (2021): `sketch`, `stylized`, `edge`, `silhouette`, `cue conflict`. Images from these domains are originally from ImageNet, but have had their properties modified. E.g., `sketch` contains ImageNet images converted to their sketch form, `stylized` contains images which have their content preserved, but with different style.

Results: Fig. 5 (right) shows the consensus scores between the teacher and the student (y-axis). We see a nearly consistent increase in consensus brought about by the distillation methods in both settings. The extent of this increase, however, differs among the domains. There are cases, for example, in VGG19 $\rightarrow$ ResNet18, where consensus over images from `edge` domain increases over 100% (12% $\rightarrow \geq$30%) by some distillation methods. On the other hand, the increase can also be relatively modest in certain cases: e.g. Swin-Base $\rightarrow$ Swin-Tiny in `stylized` domain.

Discussion: Stanton et al. (2021) showed that the agreement in classification (regardless of whether the classification itself is correct or not) between the teacher and distilled student is not much different than that to an independent student on CIFAR. We find this to be true in our ImageNet classification experiments as well. That is why the increase in agreement observed for unseen domains is surprising. After all, if the agreement is not increasing when images are from the seen domain, why should it increase when they are from an unseen domain? A possible explanation is that there is more scope for increase in agreement in unseen domains vs seen domain. The consensus score between teacher and independent student for the seen domain is $\geq$75% (Table 7), whereas for an unseen domain, e.g. `sketch`, it is $\leq$40%. So, it might not be that knowledge distillation does not work, as the authors wondered in Stanton et al. (2021), but its effect could be more prominent in certain situations.

4.5 Other studies

We study additional aspects of a model such as shape/texture bias (C.3), invariance to random crops (C.5), and find that even these obscure properties can transfer from a teacher to the student. We also explore the following questions: If we can find alternative ways of increasing a student's performance (e.g. using crafted soft labels), will that student gain similar knowledge as a distilled student (C.2)? If distillation cannot increase the student's performance, is there no knowledge transferred (C.1)?

5 Why does knowledge distillation work in this way?

We hypothesize the following: when mimicking the teacher at a particular layer, the student's intermediate representations before that layer become similar as well. That is, rather than predicting

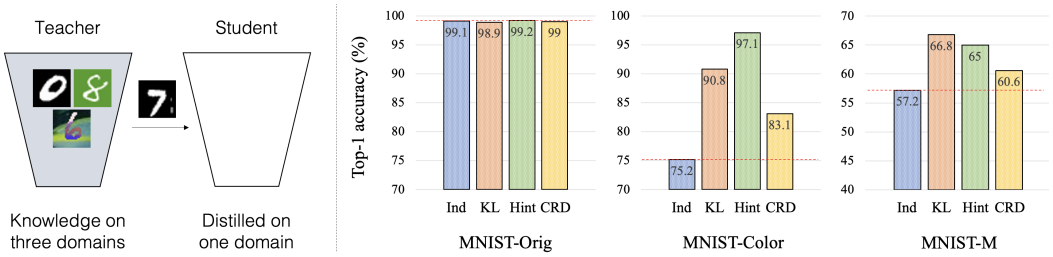


Figure 6: **Left:** The teacher is trained on three domains: `MNIST-Orig`, `MNIST-Color`, and `MNIST-M`. Distillation is done only on `MNIST-Orig`. **Right:** Test accuracy of the students. Note the increase in performance on `MNIST-Color` & `MNIST-M` domains by the distilled students.

the activations in the target layer (e.g., output layer) in a very different way (e.g., the student classifying an image based on color features while the teacher classifies it based on shape), the student learns to behave more like the teacher throughout its network. However, the degree to which this happens depends both on which layer the student mimics, and how similar the student's architecture is to that of the teacher. To study these aspects, we use centered kernel alignment (CKA) (Kornblith et al., 2019), a popular method for measuring the similarity of two neural networks. Given two representations, $X \in \mathbb{R}^{n \times p_1}$ and $Y \in \mathbb{R}^{n \times p_2}$ of the same n inputs, CKA(X, Y) $\in$ [0, 1] indicates how similar (close to 1) or dissimilar (close to 0) the two are.

Experimental setup: We consider three settings: (i) ResNet50 $\rightarrow$ ResNet18 using KD; (ii) ResNet50 $\rightarrow$ ResNet18 using $Hint$ (distillation after the default second convolutional stage); and (iii) Swin-tiny $\rightarrow$ ResNet18 using KD. For each setting, we consider representations from (roughly) corresponding locations in the network (e.g., after the last layer in each convolutional stage). Seven corresponding locations are chosen from the teacher and student (for ResNets, the same layers used in the $Hint$ ablation study, Fig. 4(d)). We take 100 random images from the ImageNet validation set and compute their representations from those layers to construct a 7 x 7 similarity matrix. We compare the teacher to both the independent and distilled student to get two similarity matrices.

Results: Figure 8 in the Appendix shows the similarities between the teacher and the independent/distilled students. First, we see that the scores are higher between the corresponding feature representations (along the diagonal entries) of the distilled student and teacher networks for ResNet50 $\rightarrow$ ResNet18, with KD resulting in a more significant gain than $Hint$. Second, we see very similar and low overall scores (except for the target F7 layer) for the independent and distilled students for Swin-tiny $\rightarrow$ ResNet18. These support our hypothesis that the student learns similar intermediate representations as the teacher before the target layer, if the student and teacher's architectures are of the same family (e.g., both are ResNets). Moreover, mimicking the output class probabilities (KD) leads to the student learning more similar representations as those of the teacher than mimicking an earlier layer ($Hint$). Finally, when the architectures are very different (Swin-tiny and ResNet18), the intermediate representations do not become similar (despite a performance gain of the distilled student) because their inductive biases lead to different ways of learning the task. Overall, our analysis shows that there is a correlation between the degree to which a student inherits the teacher's general properties and learned representation similarities.

6 Applications

Previous sections have had an exploratory angle to them, where we have primarily been interested if certain (often obscure) properties can transfer via the distillation process. Our finding, which is that the student becomes similar to the teacher on a much broader level, we believe, can have practical consequences. In this section, we present two of those - a good example and a bad example.

6.1 The Good: the free ability of domain adaptation

Consider the following setup: the teacher is trained for a task by observing data from multiple domains ($\mathcal{D}_1 \cup \mathcal{D}_2$). It is then used to distill knowledge into a student on only $\mathcal{D}_1$. Apart from getting knowledge about $\mathcal{D}_1$, will the student also get knowledge about $\mathcal{D}_2$ indirectly?

Experimental setup: We use MNIST digit recognition, and train the teacher on three domains: (i) `MNIST-orig`: original gray-scale images from MNIST, (ii) `MNIST-Color`: background of each image randomly colored, and (iii) `MNIST-M` (Ganin et al., 2016): MNIST digits pasted on random natural image patches. The student models are trained *only* on `MNIST-orig` and evaluated (top-1 accuracy) on all three domains. The network architecture is same for both the teacher and the student, consisting of two convolutional layers followed by two fully connected layers.

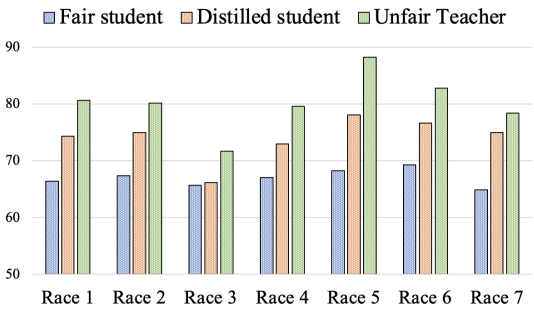
Results: When the independent student is trained only on `MNIST-orig`, its performance drops on the other unseen domains, which is expected due to domain shift. The distilled students, however, are able to significantly improve their performance on both `MNIST-Color` and `MNIST-M`; see Fig. 6.

Discussion: This result shows the practical benefits of the distillation process: once a teacher acquires an ability through computationally intensive training (e.g., training on multiple datasets), that ability can be distilled into a student, to a decent extent, through a much simpler process (access to just a single domain). The student is being shown the teacher's response to *gray-scale* images (`MNIST-orig`) that lack any color information. But somehow that information helps the student to deal better with *colored* images (`MNIST-Color` and `MNIST-M`), likely because the teacher has learned a domain-invariant representation (e.g., shape) which is distilled to the student.

6.2 The Bad: Students can inherit harmful biases from the teacher

Consider the problem of classifying gender from human faces. Imagine an independent student which performs the classification fairly across all races. The teacher, on the other hand, is biased against certain races, but is more accurate than the student *on average*. Will the student, which was originally fair, become unfair after mimicking the unfair teacher?

Experimental setup: We consider a ResNet20 $\rightarrow$ ResNet20 setting, and use FairFace dataset (Karkkainen & Joo, 2021), which contains images of human faces from 7 different races with their gender labeled. From its training split, we create two different subsets ($\mathcal{D}_s$ and $\mathcal{D}_t$) with the following objectives - (i) $\mathcal{D}_s$ has a particular racial composition so that a model trained on it will perform fairly across all races during test time; (ii) $\mathcal{D}_t$'s composition is intended to make the model perform unfairly for certain races. The exact composition of $\mathcal{D}_s$ and $\mathcal{D}_t$ is explained in Table 10. The teacher is trained on $\mathcal{D}_t$ whereas the independent/distilled students are trained on $\mathcal{D}_s$. We use KL for distillation.



Results: We start with the observation in the figure on the right that the independent (fair) student performs roughly fairly across all races, which is what we intended. The teacher is more accurate than the student. However, it performs relatively poorly on faces from Race 3 compared to other races. After distillation, we observe two things: (i) the student's *overall* accuracy improves, (ii) the gain in accuracy (compare blue and orange columns) is less for images from Race 3. To put it simply, when the student is trained on $\mathcal{D}_s$ by itself, it behaves fairly. But when it mimics the teacher on $\mathcal{D}_s$, it becomes unfair.

Discussion: The practical takeaway from the experiment is that knowledge distillation can bring forth behaviour which is considered socially problematic if it is viewed simply as a blackbox tool to increase a student's performance on test data, as it did in the previous example. Proper care must hence be taken regarding the transfer/amplification of unwanted biases in the student.

7 Conclusion

Knowledge distillation is a beautiful concept, but its success i.e, increase in student's accuracy, has often been explained by a transfer of *dark knowledge* from the teacher to the student. In this work, we have tried to shed some light on this dark knowledge. There are, however, additional open questions, which we did not tackle in this work: given the architectures of the teacher and student, is there a limit on how much knowledge can be transferred? If one wants to actively avoid transferring a certain property of the teacher into a student (Sec. 6.2), but wants to distill other useful properties, can we design an algorithm tailored for that? We hope this work also motivates other forms of investigation for us to have an even better understanding of the distillation process.

REFERENCES

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *arXiv*, 2021.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, 2006.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *arXiv*, 2020.

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *ICML*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In *NeurIPS*, 2021.

Ian Goodfellow, Jon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2014.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. In *arXiv*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Dark knowledge. *TTIC Distinguished lecture series*, 2014a.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning Workshop*, 2014b.

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. In *arXiv*, 2017.

Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *WACV*, 2021.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

Alexey Kurakin, Ian Goodfellow, and Sammy Bengio. Adversarial machine learning at scale. In *arXiv*, 2016.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *NeurIPS*, 2019.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Shunfeng Zhou, Yichao Wu, Yu Liu, Dongsheng Li, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *ICCV*, 2019.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *NeurIPS*, 2021.

Adriana Romero, Nicholas Ballas, Samira Ebrahimi Kahau, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IJCV*, 2019.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In *NeurIPS*, 2021.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.

APPENDIX

This document provides additional information complementing the main paper. First, we describe details pertaining to different distillation procedures used in Sec. A. Then, in Sec. B, we detail the iterative FGSM Kurakin et al. (2016) used to create adversarial images. Following that, in Sec. C, we perform more analyses to further dissect the distillation process, which corroborates our findings presented in the main paper. Finally, we present the top-1 accuracy of all the models, as well as the results shown in the main paper with their error bars, in Sec. D. Additionally, we have provided scripts used for evaluation performed in Sec. 4.2 and 4.3; please see `readme.txt`.

A TRAINING DETAILS

ImageNet experiments: We first describe the hyper-parameters used for different distillation objectives.

- ResNet50 $\rightarrow$ ResNet18:
  - KL: $\gamma = 0.5$, $\alpha = 0.5$
  - Hint: $\gamma = 1.0$, $\beta = 5.0$
  - CRD: $\gamma = 1.0$, $\beta = 0.8$
- VGG19 $\rightarrow$ VGG11:
  - KL: $\gamma = 1.0$, $\alpha = 0.2$
  - Hint: $\gamma = 1$, $\beta = 0.5$
  - CRD: $\gamma = 1$, $\beta = 0.8$
- VGG19 $\rightarrow$ ResNet18:
  - KL: $\gamma = 0.9$, $\alpha = 0.1$
  - Hint: $\gamma = 1$, $\beta = 0.2$
  - CRD: $\gamma = 1$, $\beta = 1.2$
- ViT $\rightarrow$ ResNet18:
  - KL: $\gamma = 1.0$, $\alpha = 0.2$
  - Hint: $\gamma = 1$, $\beta = 1$
  - CRD: $\gamma = 1$, $\beta = 0.2$
- Swin-Base $\rightarrow$ Swin-Tiny:
  - KL: $\gamma = 0.1$, $\alpha = 0.9$
  - Hint: $\gamma = 1$, $\beta = 1$
  - CRD: $\gamma = 1$, $\beta = 0.8$
- ResNet50 (sty) $\rightarrow$ ResNet18:
  - KL (lower): $\gamma = 0.1$, $\alpha = 0.9$
  - KL (higher): $\gamma = 0.9$, $\alpha = 0.1$
  - Hint (lower): $\gamma = 1.0$, $\beta = 0.2$
  - Hint (higher): $\gamma = 1.0$, $\beta = 100.0$
  - CRD (lower): $\gamma = 1.0$, $\beta = 0.8$
  - CRD (higher): $\gamma = 1.0$, $\beta = 1.2$
- ResNet50 (col) $\rightarrow$ ResNet18:
  - KL: $\gamma = 0.5$, $\alpha = 0.5$
  - Hint: $\gamma = 1.0$, $\beta = 5.0$
  - CRD: $\gamma = 1.0$, $\beta = 0.8$
- ResNet50 $\rightarrow$ ResNet18 (w/o crop):
  - KL: $\gamma = 0.5$, $\alpha = 0.5$
  - Hint: $\gamma = 1.0$, $\beta = 0.2$
  - CRD: $\gamma = 1.0$, $\beta = 0.8$

The temperature used in KL (Eq. 1 in main paper) is set to 4, and the temperature used in CRD (Eq. 3 in main paper) is set to 0.07. For CRD, the number of negative samples (N in Eq. 3) is set to 16384. For the other details, we follow the official PyTorch recommendations for training CNN-based classification models on ImageNet.[1] We train the independent students for 90 epochs, and all the distilled students for 100 epochs on ImageNet. For teacher models, we try to use those officially provided by PyTorch, whenever available. For all CNN teachers (except for stylized Res50 which is taken from here[2]) and ViT, we take models from PyTorch torchvision model zoo.[3] For Swin transformer models, we follow the training process and pretrained models given by the authors.[4] We use one 3090 Ti for training ResNet18, and two 3090 Ti for training VGG11. Each experiment takes about 2-3 days. Four A6000 are used to train Swin-T, which takes around 5 days to train.

When performing distillation using $Hint$, we need to specify the intermediate layers at which the student will mimic the teacher. Following Romero et al. (2015), we usually choose layers in the middle for that purpose. For ResNets, we choose feature after the second residual block, which has a resolution of 28×28. For VGG11 and VGG19, we choose feature after 4th and 7th conv layer whose resolution is 56×56. For Swin, we choose the feature coming after 'stage 2' (refer to Fig3 in Liu et al. (2021)), which produces a feature of 28×28 resolution. In the case of ViT-B-32 → ResNet18, the intermediate layer for ResNet18 is chosen after the fourth residual block (right before average pooling), which produces a feature of 7×7 resolution. For ViT-B-32, we choose the last layer of the encoder backbone (right before classification head), which outputs a feature having 50 dimensions. Here, we remove the classification token feature and reshape the rest into a 7×7 representation.

Note that (i) ResNet50 (sty) denotes the ResNet50 teacher trained on Stylized ImageNet dataset, which is used in Section 4.5 in the main paper; (ii) ResNet50 (col) denotes the ResNet50 teacher trained with additional color augmentations, used in Section 4.3 (color-invariance experiment); (iii) ResNet18 (w/o crop) denotes the students trained without crop augmentations used in Section 4.3 (crop-invariance experiment). Finally, the further bifurcation in ResNet50 (sty) → ResNet18 i.e., lower vs higher, denotes the hyper-parameters used when we put a lower vs higher weight on the distillation loss component, relative to the cross-entropy loss.

MNIST experiments: The architecture of both the teacher and the student, as well as all the other training details (e.g. batch size, learning rate) is taken from the standard example given by PyTorch: `Conv(32)` → `ReLU` → `Conv(64)` → `ReLU` → `MaxPool(2)` → `dropout(0.25)` → `Linear(9216, 128)` → `ReLU` → `dropout(0.5)` → `Linear(128, 10)`.[5] The distillation specific hyper-parameters are listed below:

- KL: $\gamma = 0.1$, $\alpha = 0.9$, $\tau = 8$
- $Hint$: $\gamma = 1.0$, $\beta = 2.0$, `Conv(64)` is chosen as the intermediate layer for both the teacher and the student.
- CRD: $\gamma = 1.0$, $\beta = 0.1$, $\tau = 0.1$, no. of negative samples (N) = 32.

B PROCESS OF CREATING THE ADVERSARIAL IMAGES

In Section 4.2 of the main paper, we mentioned using Iterative-FGSM Goodfellow et al. (2014); Kurakin et al. (2016) for converting a clean image (I) to its adversarial form (I^{adv}). Here, we describe that conversion process in detail. First, we pass the clean image through the target network (to be fooled). Then we compute the gradient of the loss function with respect to the image (∇_I), and then update the image in the *opposite* way, so as to maximize the loss ($J(I, y_{true})$). The update is bounded to be within a range $[I - \epsilon, I + \epsilon]$, so that the change in the image is imperceptible. This whole process constitutes one step of FGSM, and the iterative version of this method does this for k steps ($k = 5$ in our case). The process can be depicted formally through Eq. 4, where α controls the step size:

$$I_0^{adv} = I, \quad I_{t+1}^{adv} = Clip_{I,\epsilon}\big\{I_t^{adv} + \alpha \operatorname{sign}\big(\nabla_X J(I_N^{adv}, y_{true})\big)\big\} \tag{4}$$

[1] Link can be found here.

[2] Stylized Res50 can be found here.

[3] Link can be found here.

[4] Swin training code and teacher models are taken from here.

[5] Network's architecture can be found here.

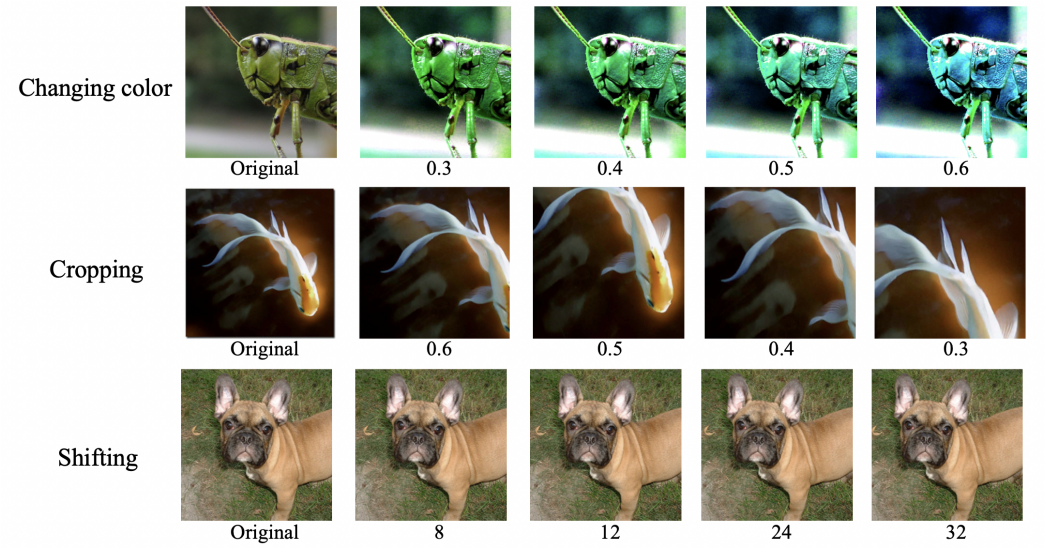


Figure 7: Visualizing the effect of data transformations. **Top:** Altering the color properties of an image (original) with increasing strengths. **Middle:** Taking random crops of an image (original) with different scale size. **Bottom:** Shifting the image left by different amounts. Color/crop invariance is studied in Sec. 4.3 of the main paper, and shift invariance is studied in Sec. C.5.

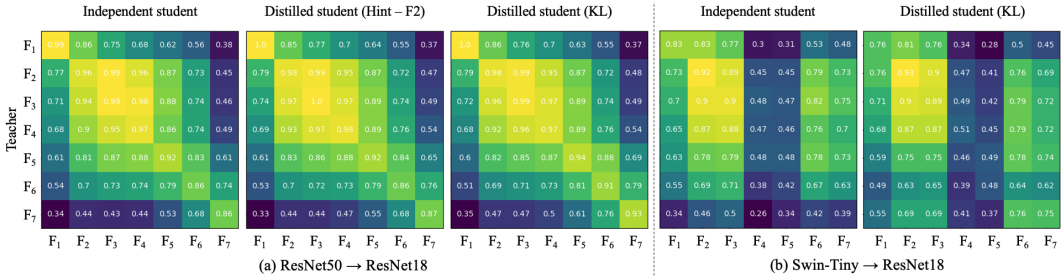
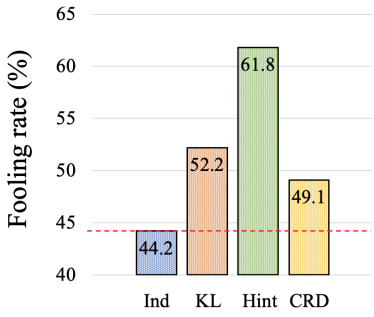


Figure 8: Centered kernel alignment (CKA) scores for various distillation settings. **Left:** Comparison of the teacher's representations with (i) the independent student, (ii) the student distilled using $Hint$ and (iii) student distilled using KL. **Right:** Comparison of the teacher (Swin-Tiny) with (i) independent student and (ii) student distilled using KL.

C More analyses

C.1 Can distillation work even without increasing student's performance?



In the experiments discussed in the main paper, the distillation objective increases the performance of the student, compared to an independent student. However, it is possible that this does not happen, as was discussed in Cho & Hariharan (2019). What do we conclude from that phenomenon? Is it that there is no knowledge transferred from the teacher to the student? In this section, we discuss such scenarios. We perform ResNet50 → ResNet18 distillation using all the distillation methods, using different hyper-parameter values (α, β, γ in Equation 1 and 2 in main paper), and choose the distilled students that are no more accurate than the independent student. The top-1 accuracy of the models are: (i) S_{Ind}: 70.03%, (ii) S_{KL}: 69.23%, (iii) S_{Hint}: 70.05% and (iv) S_{CRD}: 69.79%. Figure on the top shows the results of attacking these students using successful adversarial images crafted for ResNet50. Interestingly, the fooling rates for the distilled students are still higher compared to the independent student. So, while judging a

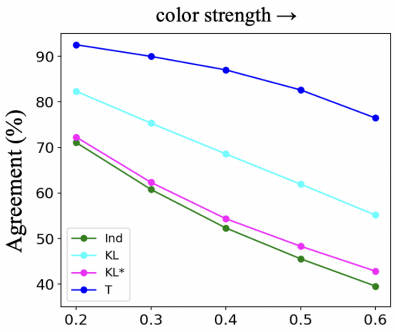
distillation setup based on the increase in student's performance is fair, it is *not* that the knowledge distillation does not work if the student's performance is not increasing.

C.2 CAN *any* SOFT LABEL TRANSFER A SIMILAR KNOWLEDGE?

When performing distillation through KL, the student has an additional target of *soft labels* from the teacher to match. In another line of work on 'label smoothing', converting the one-hot ground truth label into a softer version has also shown to improve a model's test performance Szegedy et al. (2016); Müller et al. (2019). Could this mean that using any soft label, and not necessarily obtained through a teacher, can change a student's property e.g., color invariance to the same extent?

Experimental setup: We use ResNet18 as the student and train it for ImageNet classification using KL method. However, for each input image x, instead of $\mathbf{z}_t$ (eq. 1, main paper) coming from an actual teacher, we generate the soft probabilities using x's ground-truth label $\mathbf{y}$. We first add a random Gaussian noise with variance 0.2, and then perform the softmax operation with temperature 0.15 to convert it into a probability distribution. This probability vector then acts as the target for the student to match. We then evaluate the agreement score of this *pseudo*-distilled student for color invariance (similar to Figure 4(b) in main paper).

Results: We discuss three models, (i) the independent student (Ind): top-1 acc. = 70.04%, (ii) student distilled using color-invariant ResNet50 as the teacher (KL): top-1 acc. = 71.10%, and (iii) student distilled through the soft-labels without the teacher (KL^*): top-1 acc. = 70.49%. In the figure on the right, we see that while using soft-labels does marginally increase the agreement score of the student, it does not match the scores obtained by the students distilled with the actual color-invariant teacher. This reinforces the observation we made in section 4.3, that an increase in color invariance *is primarily* due to certain knowledge being inherited from the teacher.



C.3 DOES SHAPE/TEXTURE BIAS GET DISTILLED?

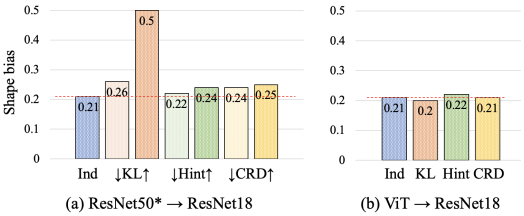
The previous section dealt with knowledge about images from unseen domains, and the section before that discussed if certain invariances can be transferred. This section brings together those ideas to study an important property: shape/texture bias of neural networks. Prior work has shown that convolutional networks tend to overly rely on texture cues when categorizing images Geirhos et al. (2019). Here we study the following: If the teacher is shape biased, and the default (independent) student more texture biased, does distillation increase the shape bias of the distilled student?

Experimental setup: We use the toolbox in Geirhos et al. (2021) to compute the shape vs. texture biases of a model. Shape bias is computed by using images with conflicting content and style information: e.g., an image with a shape (content) of a *cat* but texture (style) of an *elephant*. So, this particular image could have two correct decisions, a *cat* or an *elephant*. Using such images, the task is to see what fraction of correct decisions are based on shape vs. texture information. For the teacher, we choose a ResNet50* trained on Stylized-ImageNet Geirhos et al. (2019), where the image labels are kept the same, but the style is borrowed from arbitrary paintings. This way, the teacher has to focus more on shape information and consequently has a high shape bias of ∼0.81. We choose ResNet18 as the student, as it has a lower shape bias of ∼0.21. We then perform ResNet50* → ResNet18 distillation on the standard ImageNet dataset; i.e., the student is trained without any stylized images, while the teacher is, and we evaluate whether the student inherits the shape bias of the teacher. We also conduct an experiment with a transformer teacher and CNN student: ViT → ResNet18. Since ViT have been shown to be inherently more shape-biased, we do not train the ViT teacher on Stylized-ImageNet, and instead train both it and the student on standard ImageNet.

Results: For each distillation method, we show two results: one with lower weight on the distillation loss (↓) and one with higher (↑). From (a) in the right figure, we see that both KL and CRD improve the distilled student's shape bias, with a further jump obtained when using a higher weight, especially through KL. Sec. 4.3 already showed that the student can indirectly inherit shift/color invariance

properties of the teacher. But, it is still interesting to see that, with proper hyperparameters, the inherited knowledge includes more subtle properties, like *texture invariance* as well.

For ViT (shape bias = 0.615) → ResNet18, the shape bias of the distilled students do not change much (b). This follows a general trend where distilling knowledge from a transformer into a CNN turns out to be difficult. The implicit biases introduced due to architectural differences between the teacher and student, seem too big to be overcome by current distillation methods.



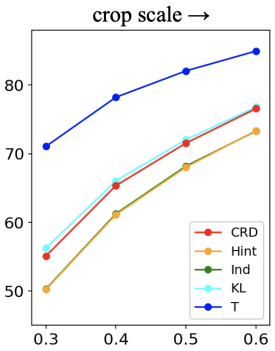
(a) ResNet50* → ResNet18 (b) ViT → ResNet18

C.4 Does invariance to random crops transfer during knowledge distillation?

This section extends the study done in Sec. 4.3 but for another popular data augmentation technique: randomly resized crops.

Experimental setup (crop invariance): While training the teacher, we randomly crop the images as part of data augmentation (in addition to horizontal flips), with crop size between 8% to 100% of the image size. So, for example, the teacher can get to see a random 20% region of an image in one iteration, and a random 80% region of the same image in a different iteration. While training the students (independent or distilled), apart from horizontal flips, we only use center crop and *do not* show random crops of an image.

Results (crop invariance): During evaluation, we start with a test image X from the 50k val set. We then set a crop scale, e.g. 0.2, and generate two random crops X_1 and X_2 so that both cover a random 20% area of the original image X. Higher the crop scale, more image content will be common between the two crops. Then, we measure how frequently a model assigns the same class to X_1 and X_2. Fig. 4 (d) shows the agreement scores for increasing crop scales, where we again observe that the students distilled through KL and CRD become more invariant to this operation. Student distilled through $Hint$, however, does not increase its invariance to random crops, just as it did not increase its invariance to color jittering to the same extent as other methods in Fig. 4(b).



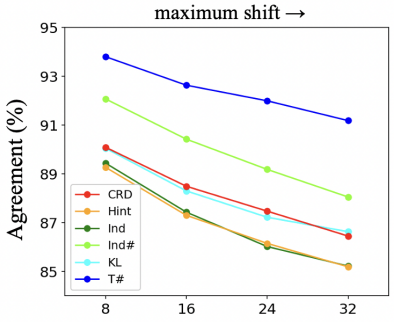
C.5 Does shift invariance transfer during knowledge distillation?

Section 4.3 and C.4 discussed whether invariance to certain data transformations can transfer from a teacher to the student during knowledge distillation. Fig. 7 visualizes the effect of those transformations. Note that when we generate two random crops (X_1, X_2) of an image (X) with a fixed scale (e.g. 0.4), the aspect ratio of the two crops can still be kept different, which is what we do in Fig. 7 (middle) and in the results shown in the previous section. If the aspect ratio is kept the same between X_1 and X_2, then one can study a more common property of neural networks: *shift invariance* i.e. whether the network's predictions remain same if we shift an image by certain pixels (either left/right/top/bottom). We study if this knowledge can be transferred from a teacher to the student during the distillation process.

Experimental setup: For the teacher, we choose a model which has been explicitly made to be shift-invariant. A recent work showed that a model's robustness to input shifts is related with the aliasing phenomenon, which refers to signal distorted with a small downsampling rate. To alleviate this issue and make CNNs shift invariant, Zhang (2019) inserts low-pass filters into CNNs before downsampling. So, we use an anti-aliased ResNet50# as the teacher (# represents anti-aliased, same for the below). The student is the standard ResNet18 (without being anti-aliased). The distillation ResNet50# → ResNet18 is done on the standard ImageNet dataset. The shift invariance of a model is evaluated across the 50k validation images in ImageNet. We start with a test image X resized into 256x256 resolution. Then, we define the maximum shift we want in the resulting two images. If, for example, that value is 32, then we do a center crop of 256x256 followed by two random

224x224 crops to generate X_1 and X_2, keeping the aspect ratio same for both. If, instead, we desire a maximum shift of only 8 between X_1 and X_2, we would do a center crop of 232x232, followed by by two random 224x224 crops. Then, we compute how frequently a model gives the same prediction for X_1 and X_2, which is called the agreement score (same as section 4.3).

Results: In the figure on the right, we see the agreement scores of different models, and see that the agreement scores of the ResNet18 students distilled using KL and CRD increase relative to the independent ResNet18. Note that one can convert ResNet18 (the student) into its anti-aliased version as well by inserting low-pass filters Zhang (2019). The agreement score achieved by this student can be thought of as the upper-limit for a ResNet18 model, which we show by light green colored plot (denoted as Ind#). Given the results of section 4.3 (crop-invariance), this result is expected since invariance to image shifts (aspect ratio constant) is a subset of invariance to random crops (aspect ratio could be different). Again, we observe that $Hint$ has difficulty in transferring this property.



D SUPPORTING QUANTITATIVE RESULTS

Finally, we report the performance of different models on ImageNet 50k validation set. Table 1 lists the top-1 accuracies of different models used in the main paper. Overall, we have tried to use the hyper-parameters which improve the distilled student's performance compared to the independent student. In every case, we use a single teacher to perform distillation into two students trained with different random seeds i.e. Teacher $\rightarrow$ Student$_1$ and Teacher $\rightarrow$ Student$_2$, for each method. We then report the results shown in the main paper with their respective error bars, in Tables 2-9.

	Teacher	**Ind**	**KL**	**Hint**	**CRD**
ResNet50 $\rightarrow$ ResNet18	76.13	70.04±0.01	70.98±0.01	70.56±0.16	70.73±0.02
VGG19 $\rightarrow$ VGG11	72.37	68.88±0.01	69.74±0.10	69.38±0.15	69.74±0.07
VGG19 $\rightarrow$ ResNet18	72.37	70.04±0.01	70.62±0.02	70.21±0.30	70.42±0.07
ViT $\rightarrow$ ResNet18	75.91	70.04±0.01	70.39±0.02	70.59±0.07	70.58±0.03
Swin-Base $\rightarrow$ Swin-Tiny	83.50	81.13±0.08	81.23±0.04	81.33±0.11	81.27 ±0.21
ResNet50 (sty) $\rightarrow$ ResNet18 $\uparrow$	60.18	70.04±0.01	61.45±0.07	68.82±0.12	69.56±0.07
ResNet50 (sty) $\rightarrow$ ResNet18 $\downarrow$	60.18	70.04±0.01	70.65±0.03	70.45±0.07	69.96±0.05
ResNet50 (col) $\rightarrow$ ResNet18	75.32	70.04±0.01	71.01±0.06	70.41±0.20	70.97±0.19
ResNet50 $\rightarrow$ ResNet18 (w/o crop)	76.13	64.84±0.02	68.75±0.01	64.81±0.14	67.41±0.07

Table 1: Top-1 accuracy (in %) of different models on 50k ImageNet validation images.

	Teacher	**Ind**	**KL**	**Hint**	**CRD**
ResNet50 $\rightarrow$ ResNet18	84.82	44.16±0.19	51.98±2.44	48.34±0.34	50.46±0.29
VGG19 $\rightarrow$ VGG11	87.22	62.29±0.36	69.74±0.67	79.78±0.08	70.51±0.78
VGG19 $\rightarrow$ VGG11 (R18)	87.22	69.02±0.48	70.54±0.90	70.68±0.62	70.59±0.62
ViT $\rightarrow$ ResNet18	85.84	21.93±0.24	21.57±0.49	23.34±0.14	23.47±0.31
VGG19 $\rightarrow$ ResNet18	87.22	36.19±0.01	43.02±0.06	47.68±0.47	48.99±0.05

Table 2: Adversarial fooling rates (in %), corresponding to Figure 3 in the main paper.

	ResNet50 (col)	**ResNet50**
Ind	71.27±0.21	71.27±0.21
KL	82.10±0.07	74.02±0.23
Hint	72.22±0.14	72.42±0.39
CRD	79.44±0.20	71.27±0.25

Table 3: Table corresponding to Figure 4(a) in main paper. Knowledge transfer about color information from two teachers: color invariant ResNet50 (T) and default ResNet50 (T*).

	0.3	**0.4**	**0.5**	**0.6**
Ind	60.77±0.10	52.32±0.21	45.55±0.20	39.56±0.31
KL	75.32±0.17	68.56±0.36	61.93±0.33	55.15±0.37
Hint	61.96±0.00	53.34±0.41	47.72±0.51	42.00±0.53
CRD	71.83±0.48	64.31±0.14	57.26±0.47	49.90±0.48

Table 4: Table corresponding to Figure 4(b) in main paper. Illustration of knowledge transfer in, ResNet50 → ResNet18, if the two images have increasingly different color properties.

	0.3	**0.4**	**0.5**	**0.6**
Ind	60.77±0.10	52.32±0.21	45.55±0.20	39.56±0.31
KL	62.49±0.09	54.18±0.15	47.68±0.50	41.92±0.48
Hint	61.68±0.04	53.63±0.19	47.37±0.32	41.76±0.50
CRD	60.85±0.65	52.80±0.76	46.91±1.29	42.00±1.31

Table 5: Table corresponding to Figure 4(c) in main paper. Illustration of knowledge transfer in, Swin-Tiny → ResNet18, if the two images have increasingly different color properties.

	0.3	**0.4**	**0.5**	**0.6**
Ind	50.30±0.21	61.27±0.10	68.20±0.50	73.30±0.33
KL	56.26±0.00	66.06±0.02	72.06±0.19	76.79±0.04
Hint	50.23±0.27	61.14±0.11	68.04±0.14	73.36±0.08
CRD	55.10±0.12	65.36±0.43	71.55±0.26	76.57±0.43

Table 6: Table corresponding to Figure 4(d) in main paper. Illustration of knowledge transfer in, ResNet50 → ResNet18, if the two images are random crops of increasing scales.

	VGG19 → ResNet18					
	sketch	**stylized**	**silhouette**	**edge**	**cue conflict**	**ImageNet val**
Ind	33.62±0.12	21.68±0.19	12.81±0.94	26.25±1.25	22.81±0.00	75.60±0.01
KL	37.56±0.31	28.81±0.44	31.25±5.00	31.25±5.00	29.30±2.03	77.21±0.06
Hint	37.18±1.19	27.19±0.19	10.00±3.75	29.69±2.19	27.73±1.25	76.49±0.09
CRD	40.50±0.37	30.75±0.50	37.81±2.19	35.00±1.25	30.93±0.08	78.36±0.06

	Swin-Base → Swin-Tiny					
	sketch	**stylized**	**silhouette**	**edge**	**cue conflict**	**ImageNet val**
Ind	51.37±0.37	33.75±0.37	22.50±1.25	50.00±0.00	37.26±0.47	88.79±0.07
KL	56.93±1.06	38.43±0.68	27.50±2.50	57.81±1.56	42.61±0.04	89.39±0.05
Hint	52.56±1.18	35.18±0.19	26.87±1.87	54.37±2.50	38.51±0.62	89.03±0.17
CRD	54.18±0.94	34.75±1.50	26.25±0.00	50.62±0.00	39.22±0.47	88.97±0.01

Table 7: Consensus scores between teacher and the student, corresponding to Figure 5 in the paper. ImageNet val denotes the 50k images in the validation set of the seen domain (ImageNet).

	ResNet50 (sty) → ResNet18		ViT → ResNet18
	Lower	Higher	
Ind	0.21±0.01	0.21±0.01	0.21±0.01
KL	0.26±0.01	0.50±0.00	0.20±0.01
Hint	0.22±0.01	0.24±0.02	0.22±0.00
CRD	0.24±0.00	0.25±0.01	0.21±0.00

Table 8: Shape bias scores of students, corresponding to the figure in Section 4.5 in the main paper.

	MNIST-orig	MNIST-Color	MNIST-M
Ind	99.08±0.07	72.86±2.32	56.09±1.14
KL	98.90±0.01	91.76±1.00	67.92±1.07
Hint	99.10±0.06	97.05±0.05	64.06±0.93
CRD	99.00±0.10	83.98±0.88	60.36±0.23

Table 9: Top-1 accuracy of distilled models, corresponding to figure 6 in the main paper.

	$\mathcal{D}_s$	$\mathcal{D}_t$
Race 1	600	4000
Race 2	50	4000
Race 3	2000	0
Race 4	200	4000
Race 5	0	4000
Race 6	200	4000
Race 7	800	4000

Table 10: Dataset composition of FairFace (Karkkainen & Joo, 2021). Different rows represent the number of training images used from each race.