Appendix

This document provides additional information complementing the main paper. First, we describe details pertaining to different distillation procedures used in Sec. 1. Then, in Sec. 2, we detail the iterative FGSM [7] used to create adversarial images. Following that, in Sec. 3, 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. 4. Additionally, we have provided scripts used for evaluation performed in Sec. 4.2 and 4.3; please see `readme.txt`.

1 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 [11], 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 [9]), 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.

2 Process of creating the adversarial images

In Section 4.2 of the main paper, we mentioned using Iterative-FGSM [4, 7] 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. 1, where α controls the step size:

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

[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.

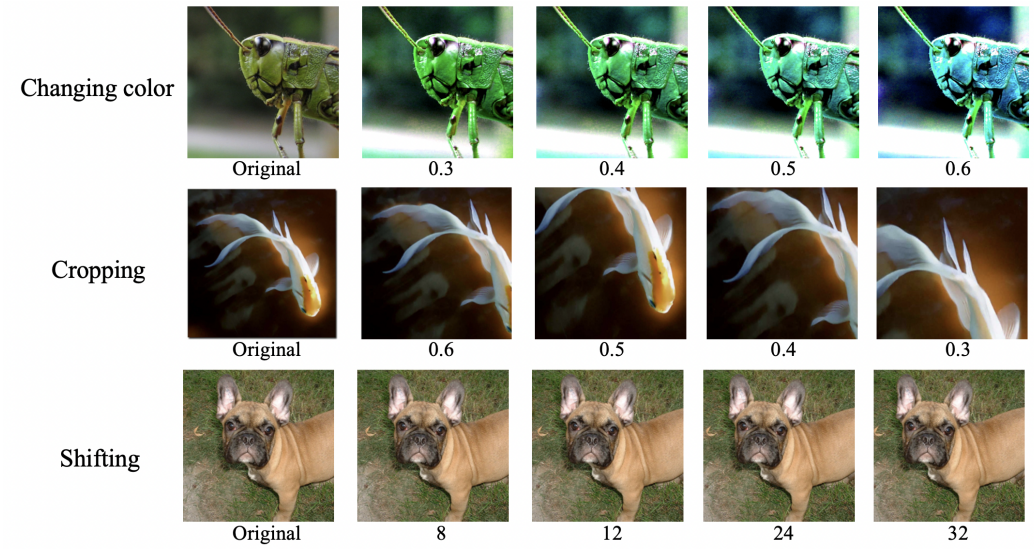


Figure 1: Visualizing the effect of data transformations. **Top:** Altering the color properties of an image (original) with increasing strengths. **Middle:** Taking random crops of an image (original) with different scale size. **Bottom:** Shifting the image left by different amounts. Color/crop invariance is studied in Sec. 4.3 of the main paper, and shift invariance is studied in Sec. 3.4.

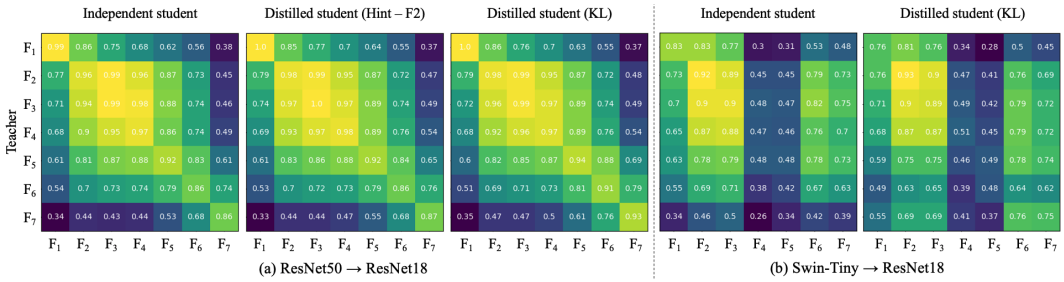
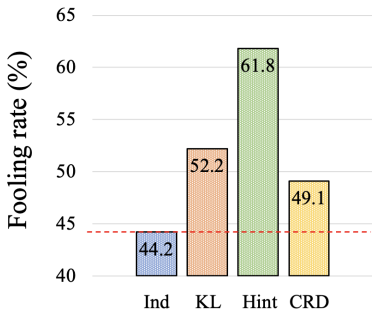


Figure 2: Centered kernel alignment (CKA) scores for various distillation settings. **Left:** Comparison of the teacher's representations with the independent and two distilled students (KL and $Hint$). **Right:** Comparison of the teacher (Swin-Tiny) with independent and distilled student (KL).

3 More analyses

3.1 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 [1]. 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 $\rightarrow$ 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.

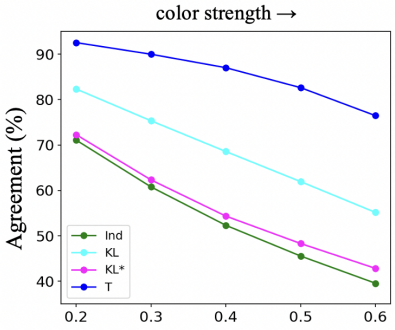


3.2 Can *any* soft label transfer a similar knowledge?

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 [12, 10, 8, 14]. 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.

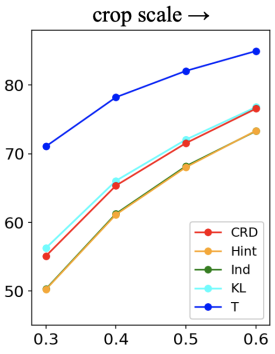


3.3 Does invariance to random crops transfer during knowledge distillation?

This section extends the study done in Sec. 4 of the main paper, 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) (main paper) 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) (main paper).



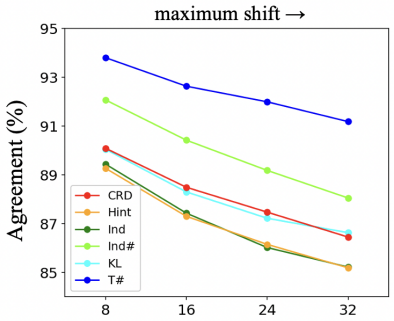
3.4 Does shift invariance transfer during knowledge distillation?

Section 4 (main paper) and 3.3 (appendix) discussed whether invariance to certain data transformations can transfer from a teacher to the student during knowledge distillation. Fig. 1 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. 1 (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, [13] 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# $\rightarrow$ 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 [13]. 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.



3.5 Does shape/texture bias get distilled?

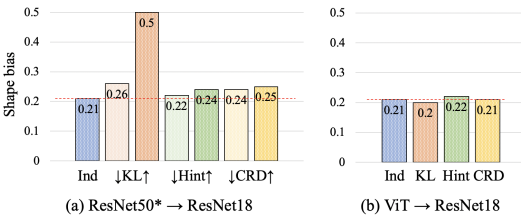
The previous section dealt with knowledge about images from unseen domains, and the section before that discussed if certain invariances can be transferred. This section brings together those ideas to study an important property: shape/texture bias of neural networks. Prior work has shown that convolutional networks tend to overly rely on texture cues when categorizing images [3]. 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 [2] 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 [3], 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 $\sim$0.81. We choose ResNet18 as the student, as it has a lower shape bias of $\sim$0.21. We then perform ResNet50* $\rightarrow$ 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 $\rightarrow$ 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 ($\downarrow$) and one with higher ($\uparrow$). 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 (main paper) already showed that the student can indirectly inherit 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) $\rightarrow$ 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* $\rightarrow$ ResNet18 (b) ViT $\rightarrow$ ResNet18

3.6 Distillation makes internal representations to become similar

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 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) [6], 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. 4d). 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 2 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.

4 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 → ResNet18	76.13	70.04±0.01	70.98±0.01	70.56±0.16	70.73±0.02
VGG19 → VGG11	72.37	68.88±0.01	69.74±0.10	69.38±0.15	69.74±0.07
VGG19 → ResNet18	72.37	70.04±0.01	70.62±0.02	70.21±0.30	70.42±0.07
ViT → ResNet18	75.91	70.04±0.01	70.39±0.02	70.59±0.07	70.58±0.03
Swin-Base → Swin-Tiny	83.50	81.13±0.08	81.23±0.04	81.33±0.11	81.27 ±0.21
ResNet50 (sty) → ResNet18 ↑	60.18	70.04±0.01	61.45±0.07	68.82±0.12	69.56±0.07
ResNet50 (sty) → ResNet18 ↓	60.18	70.04±0.01	70.65±0.03	70.45±0.07	69.96±0.05
ResNet50 (col) → ResNet18	75.32	70.04±0.01	71.01±0.06	70.41±0.20	70.97±0.19
ResNet50 → 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 → ResNet18	84.82	44.16±0.19	51.98±2.44	48.34±0.34	50.46±0.29
VGG19 → VGG11	87.22	62.29±0.36	69.74±0.67	79.78±0.08	70.51±0.78
VGG19 → VGG11 (R18)	87.22	69.02±0.48	70.54±0.90	70.68±0.62	70.59±0.62
ViT → ResNet18	85.84	21.93±0.24	21.57±0.49	23.34±0.14	23.47±0.31
VGG19 → 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.

References

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

	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.

[2] 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.

[3] 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.

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

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

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

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

[8] Yuan Li, Francis E.H.Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, 2020.

[9] 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.

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

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

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

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

	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 [5]. Different rows represent the number of training images used from each race.

[14] Zhilu Zhang and Mert Sabuncu. Self-distillation as instance-specific label smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2184–2195. Curran Associates, Inc., 2020.