What do CNNs Learn in the First Layer and Why? A Linear Systems Perspective

Rhea Chowers [1] **Yair Weiss** [1]

Abstract

It has previously been reported that the representation that is learned in the first layer of deep Convolutional Neural Networks (CNNs) is highly consistent across initializations and architectures. In this work, we quantify this consistency by considering the first layer as a filter bank and measuring its energy distribution. We find that the energy distribution is very different from that of the initial weights and is remarkably consistent across random initializations, datasets, architectures and even when the CNNs are trained with *random labels*. In order to explain this consistency, we derive an analytical formula for the energy profile of linear CNNs and show that this profile is mostly dictated by the second order statistics of image patches in the training set and it will approach a whitening transformation when the number of iterations goes to infinity. Finally, we show that this formula for linear CNNs also gives an excellent fit for the energy profiles learned by commonly used *nonlinear* CNNs such as ResNet and VGG, and that the first layer of these CNNs indeed performs approximate whitening of their inputs.

1. Introduction

The remarkable success of Convolutional Neural Networks (CNNs) on a wide variety of image recognition tasks is often attributed to the fact that they learn a good representation of images. Support for this view comes from the fact that very different CNNs tend to learn similar representations and that features of CNNs that are trained for one task are often useful in very different tasks (Yosinski et al., 2014; Gidaris et al., 2018a; Doimo et al., 2020).

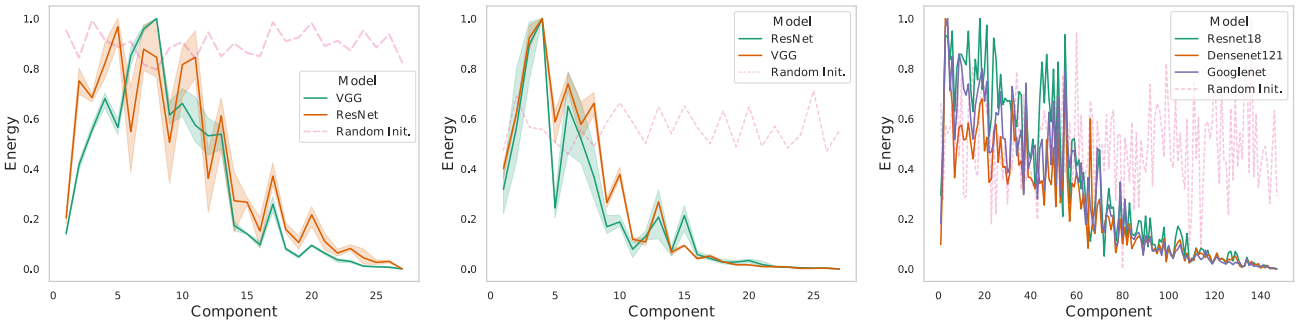
A natural starting point for investigating representation learning in deep CNNs is the very first layer. Studying this representation is somewhat easier than studying more general representation learning for the simple reason that the output of this layer is a linear function of its input. Thus we can use the perspective of linear systems whereby a system based on convolutions can be fully characterized by its frequency response. In this paper, we adopt the linear systems perspective and consider the first layer as a filter bank and measure the sensitivity of the bank to different spatial frequencies. As we show in Section 2, this profile of sensitivities (which we call the "energy profile") is highly consistent for different initializations, architectures and training sets and is very different from the profile of the initial random weights. The filter bank's sensitivity peaks at intermediate spatial frequencies, while being *insensitive* to very high and low spatial frequencies.

The linear systems perspective has been used in the past to analyze biological neural networks (Atick & Redlich, 1990) where it has been argued based on first principles that the first layer of a neural network should perform "redundancy reduction" (Barlow, 1989). For example, in the case of images, the pixel representation is highly redundant since neighboring pixel values are highly correlated. Under the redundancy reduction hypothesis, the goal of early layers is to "disentangle" the input and remove these correlations to facilitate downstream learning. When this hypothesis is formalized, the resulting optimal transformation takes the form of "whitening": the sensitivity of the first layer to a particular frequency should be inversely proportional to the variance of the input signal at that frequency (provided that the input variance is much larger than the noise). Such "whitening" transformations have been observed experimentally in different biological systems (Hyvärinen et al., 2009), and several authors have recently argued that whitening should be enforced in the different layers of CNNs (Huang et al., 2018; Zhang et al., 2021).

If CNNs were trained with an explicit "redundancy reduction" loss function, we would therefore expect their energy profiles to be consistent for different architectures and random initializations, but why does this consistency occur when the networks are trained to minimize a classification loss on the training set? A possible explanation is that these filters are optimal in some sense for solving the recognition task. Thus, the networks have simply learned that in order to minimize the training loss, the first layer of deep CNNs

[1] School of Computer Science and Engineering, Hebrew University, Jerusalem, Israel. Correspondence to: Rhea Chowers <rhea.chowers@mail.huji.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.



(a) CIFAR10 (b) CelebA (c) Imagenet

Figure 1. The energy profiles of different learned CNNs are highly consistent and different from initializaiton. CIFAR10 and CelebA are averaged over many different initializations and the spread indicates the variance. Models trained on ImageNet were downloaded from the PyTorch library. See Section 2 for full correlation coefficients. An example of a random initialization is plotted for reference.

must have filters whose energy profile has a particular shape.

In this paper we present empirical and theoretical results that are inconsistent with this explanation. We show that trained networks learn consistent representations that are far from their initialization despite the fact that CNNs with commonly used architectures can be trained equally well with frozen random filters in the first layer. We also show that the same energy profile is obtained when the network is trained to predict random labels. We then show that under realistic assumptions on the statistics of the input and labels, consistency also occurs in simple, linear CNNs, and derive an analytical form for its energy profile. We show that as the number of iterations goes to infinity, this profile takes the form of a first layer that performs whitening of the input image patches. Finally, we show that the analytical formula which we derived for linear CNNs gives an excellent fit to the energy profile of real-world CNNs as well, when trained with either true or random labels. Our code is publicly available[1].

2. Quantifying Consistency using Energy

Defining the similarity between the representations learned by different CNNs is challenging (Laakso & Cottrell, 2000; Wang et al., 2018). The dimension of the representation may be different and even when they are the same, the two representations may be very different when individual neurons are compared but still identical when the full representation is compared (e.g. two representations that are rotations of each other). Recent works (Kornblith et al., 2019; Nguyen et al., 2021) suggest comparing two representations based on the distance between the distribution over patches induced by the two representations. But estimating this distance in high dimensions is nontrivial and two very different networks might give similar distributions over patches when the input distribution is highly skewed (Ding et al., 2021). We propose a new method which avoids these shortcomings and is especially relevant for the first layer of a CNN.

Our method is based on the linear systems perspective, whereby a system that is based on convolutions is fully specified by its frequency response. Since the filters in CNNs are typically highly localized in space (e.g. many successful CNNs use $3 \times 3 \times 3$ filters in the first layer) we characterize this frequency response using the principal components of the input image patches.

Definition 2.1. Given a set of patches $\{p_n\}$ the PCA vectors u_i are eigenvectors of the matrix $\sum_n p_n p_n^T$.

Definition 2.2. Given a set of filters $\{w_k\}$ and a set of PCA vectors $\{u_i\}$ the *energy profile* of the set is given by a vector e whose i'th component is given by:

$$e_i^2 = \frac{1}{K} \sum_{k=1}^{K} (w_k^T u_i)^2 \tag{1}$$

We measure similarity between two different sets of filters by measuring the correlation coefficient between their energy profiles. Note that this measure is invariant to a rescaling of the filters, to a permutation of the filters and to any orthogonal transformation of the filters. Since the PCA vectors of a set of patches extracted from natural images are highly localized in frequency (Hyvärinen et al., 2009), this way of comparing linear representations is equivalent to considering the set of filters as a filter bank and measuring the sensitivity of the filter bank to different spatial frequencies.

Figure 1 shows that different models trained with gradient descent are remarkably consistent using our proposed measure. Regardless of architecture or the particular dataset that they were trained on, different CNNs have very similar energy profiles that are less sensitive to very high or low spatial frequencies, and the peak sensitivity is for intermediate

[1] https://github.com/RheaChowers/CNNs-First-Layer

DATASET	SEED	WIDTH	TRAINED VS INIT	VGG VS RESNET
CIFAR10	0.99	0.98	-0.13 $\pm$ 0.18	0.87
CIFAR100	0.97	0.98	-0.04 $\pm$ 0.04	0.80
CELEBA	0.99	0.98	-0.18 $\pm$ 0.13	0.92

Table 1. Correlation between energy profiles of VGG11 (Simonyan & Zisserman, 2015), trained with different random seeds (initializations), first layer widths, over various datasets, and compared with ResNet18 (He et al., 2016). Standard deviation provided in cases it exceeds 0.04.

spatial frequencies. This profile is very different from the profile of the initial, random, filters which is approximately constant for all frequencies.

Section 2 quantifies this similarity. The correlation between energy profiles of trained models with different random initializations and architecture is remarkably high (over 0.98 in the case of different seeds and first layer widths) and the correlation between the learned profiles and the random initialization is close to zero. An extensive set of comparisons of various models and datasets can be found in Appendix E.

Thus the use of our new measure allows us to quantitatively show that deep CNNs trained with gradient descent using standard parameters exhibit highly consistent representation, namely in the form of sensitivity to intermediate spatial frequencies. We now ask: what determines this consistency?

3. Is Consistency due to CNNs Learning Semantically Meaningful Features?

A natural explanation for the remarkable consistency of the learned representation in the first layer is that CNNs learn a representation that is good for object recognition. In particular, high spatial frequencies are often noisy while very low spatial frequencies are often influenced by illumination conditions. Thus learning a representation that is mostly sensitive to intermediate spatial frequencies makes sense if the goal is to recognize objects. Similarly, human vision is also mostly sensitive to intermediate spatial frequencies (Owsley, 2003), presumably for the same reasons.

In order to test this hypothesis we asked if training modern CNNs while freezing the first layer will result in a decrease in performance. If indeed a set of filters that is sensitive mostly to intermediate frequencies is optimal for object recognition, we would expect performance to suffer if we froze the first layer to have random filters with equal energy in all frequencies.

Figure 2 shows that there is almost no change in the performance of modern CNNs when the weights in the first layer are frozen. This is true when measuring training accuracy, training loss or validation accuracy and loss (see Appendix D). Apparently the networks learn to compensate for the random filters in the first layer by learning different weights in the subsequent layers. In other words, if we were to train modern CNNs using some discrete search over weights (e.g. genetic programming) to minimize the training loss, there is no reason to expect a consistent energy profile that is sensitive mostly to intermediate spatial frequencies to be found. Equally good training loss can be obtained with random filters in the first layer.

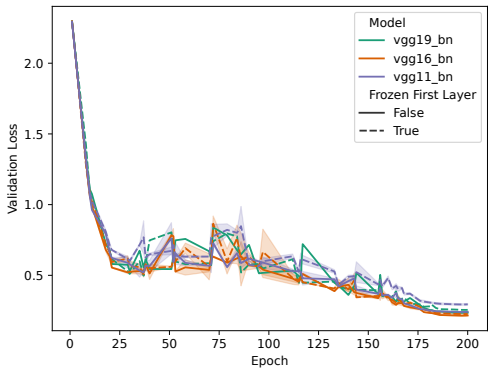
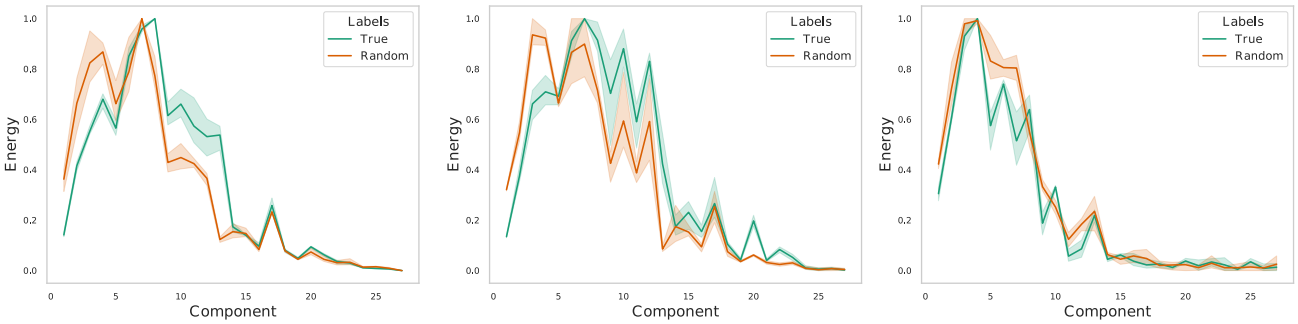


Figure 2. Validation loss of VGGs of different depths on CIFAR10 as function of iteration with frozen first layer and without. For deep networks the performance is the same as with frozen layer. Training loss and accuracy figures can be found in Appendix D

Another test to this hypothesis can be done by training networks with random labels. In this setting, models are known to memorize their training set (Arpit et al., 2017). While a particular energy profile may be optimal for recognizing natural object categories (e.g. for ignoring illumination effects), we should not expect any particular set of features to be optimal for recognizing randomly defined categories. Surprisingly, however, we find *the same energy profile when CNNs are trained with true labels and random labels.* Figure 3 compares the energy profiles of models trained with true and random labels on different datasets, and shows a highly consistent profile between the two sets of labels. Section 3 shows that this result is consistent over multiple random seeds and far from initialization.

To summarize, while quantitatively highly consistent representations are learned in the first layer of commonly used CNNs, this cannot be explained by the networks minimization of the training loss. Furthermore, the learned set of features is consistent for models trained with *random labels* as well, suggesting a bias in the input, training algorithm, or both. This motivates us to analyze representation learning in much simpler CNNs.



(a) CIFAR10 (b) CIFAR100 (c) CelebA

Figure 3. VGG11 trained on CIFAR10 (Figure 3a) CIFAR100 (Figure 3b) and a CelebA classification task (Figure 3c) exhibit similar energy patterns when trained with true and random labels. These are also highly correlated and differ from initialization (see Section 3). Further experiments on binary CIFAR10 subsets can be found in Appendix E.

DATASET	VGG (RANDOM) V INIT	VGG (TRUE) VS VGG (RANDOM)
CIFAR10	0.03 ± 0.22	$\mathbf{0.90 \pm 0.02}$
CIFAR100	0.14 ± 0.13	$\mathbf{0.91 \pm 0.01}$
CELEBA	0.08 ± 0.07	$\mathbf{0.96 \pm 0.03}$

Table 2. Correlation between energy profiles of VGG11 trained with true and random labels for different datasets. While highly correlated, the profiles are far from initialization.

4. Theory in Simple Linear CNNs

In order to understand the consistency that we observe among energy profiles in the first layer of trained CNNs, we turn to analyzing a very simple model: a *linear* CNN with one hidden layer trained with the MSE loss. Specifically, in this simple model, the first layer includes convolutions with K different filters and the output is given by a global average pool of the filters over all locations.

This model is clearly very different from real-world CNNs, but it allows a closed form analysis of the energy profile in the first layer. Furthermore, we will subsequently show that it exhibits many of the same properties as those of real-world CNNs.

Our main theorem (4.2) provides an analytic formula for the energy profile of these models, which is consistent across initializations and widths of the first layer. Additionally, given that true labels are uncorrelated with image patches, the theorem implies consistency between models trained with true and random labels as well.

The theorem relates the energy profile of the learned filters to the energy profile of the training patches, which we now define.

Definition 4.1. Given a set of patches $\{p_n\}$ and a set of PCA vectors $\{u_i\}$ the energy profile of the set is given by a vector λ whose ith component is given by:

$$\lambda_i^2 = \frac{1}{N}\sum_{n=1}^{N}(p_n^T u_i)^2 \tag{2}$$

Theorem 4.2. *Consider a depth-2 linear CNN of any width initialized with zero mean filters and variance $\sigma^2 I$ and trained with gradient descent with step size η on the MSE loss. Assume that different patches in each image are uncorrelated with each other and that the labels are uncorrelated with individual PCA components, then as the number of patches in the training set goes to infinity, the energy profile of the filters at iterations t is given by:*

$$e_i = \tilde{c} \cdot \frac{|1-(1-\eta\lambda_i^2)^t|}{\eta^2\lambda_i^2}\lambda_i + \xi_i \tag{3}$$

where λ_i is the energy profile of the training patches and ξ a random vector that depends on the initialization and whose magnitude goes to zero as $\sigma \to 0$.

Proof Sketch. The result is obtained by explicitly calculating the gradient of the MSE loss with respect to the average filter and noting that the dynamics of gradient descent can be written as scalar dynamics in PCA space and take the form of a geometric series(LeCun et al., 1991). The result also uses the assumption that the labels are uncorrelated with the PCA coefficients to obtain a formula that does not depend on the labels. Even though the labels are uncorrelated with the PCA coefficients, any finite dataset will include small, spurious correlations and the magnitude of these correlations will almost surely be proportional to λ_i. A full proof is supplied in Appendix A. □

Thus under our assumptions, the energy profile will only depend on the second-order statistics of the input patches (as described by the energy profile λ_i), the number of iterations,

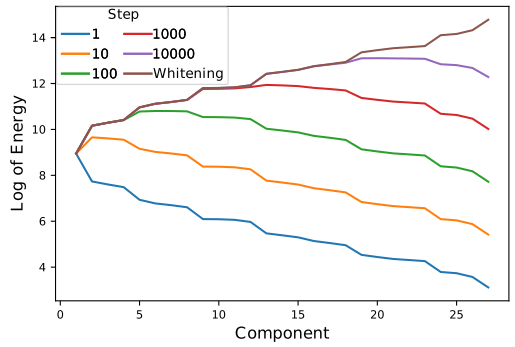


Figure 4. Energy profiles predicted by our analytic formula (Equation (4)) at different iterations with a constant learning rate for the eigenvalue spectrum of CIFAR10. At early training iterations the profile is sensitive to the largest eigenvalue (corresponding to the first PCA component). The sensitivity shifts as the number of iterations increases and at each iteration the representation performs whitening on increasingly higher frequencies.

and the learning rate. But what does this profile mean? Figure 4 shows the analytic formula at different iterations with a constant learning rate when the energy profile of the patches λ_i is calculated on CIFAR10 (note the log scale on the y axis). At early training iterations, the formula is mostly sensitive to low spatial frequencies but the sensitivity shifts as the number of iterations increases. As the number of iterations approaches infinity, the profile is actually sensitive mostly to high spatial frequencies. The following theorem shows that as the number of iterations goes to infinity, the filters of a linear CNN perform whitening.

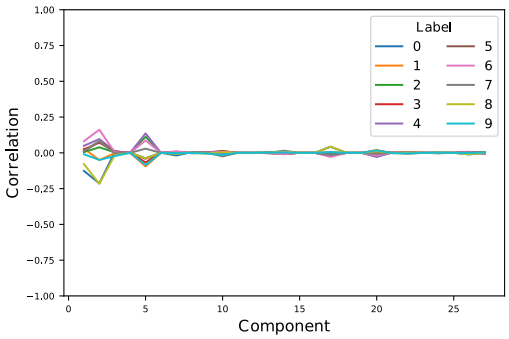
Theorem 4.3. *Let $\{w_k\}$ be the filters in the first layer of a CNN. If the energy profile of these filters satisfy:*

$$e_i = \tilde{c} \cdot \frac{|1-(1-\eta\lambda_i^2)^t|}{\eta^2\lambda_i^2}\lambda_i \tag{4}$$

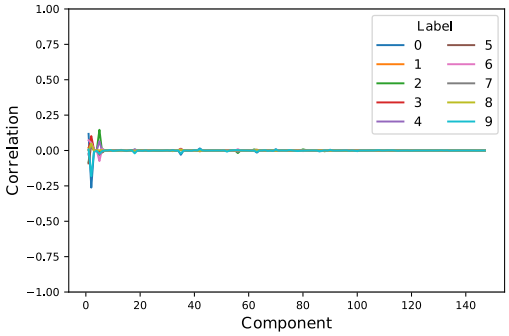
then as the number of iterations goes to infinity, the filters in the first layer of the CNN perform spatial decorrelation: the vector of responses at any given location is uncorrelated with the vector of responses at any other location.

Proof Sketch. For any learning rate $\eta < \frac{1}{\max_i \lambda_i^2}$, at the limit $t \to \infty$ then $(1-\eta\lambda_i^2)^t \to 0$, meaning $e_i \propto \frac{1}{\lambda_i}$, which is a whitening filter and therefore performs spatial decorrelation. For full proof see Appendix A. □

In other words, when assuming that the labels and input patches are uncorrelated simple linear CNNs learn consistent energy profiles which will converge to a whitening transform, i.e. a transform that performs spatial decorrelation. For finite iterations, the filters will not perform full whitening and only those components for which λ_i is large will be whitened (Figure 4). This is similar to the optimal redundancy reduction that was derived from first principles in (Atick & Redlich, 1990) and suggested that only components for which λ_i is much greater than the noise should be whitened. But unlike the explicit "redundancy reduction" discussed in previous works, here partial whitening emerges due to a trade off with the number of iterations, caused by the use of gradient descent to minimize the training loss.



(a) CIFAR10



(b) ImageNet (10)

Figure 5. Correlation between the patch energy in each PCA component and the class labels for CIFAR10 (Figure 5a, using $3\times3\times3$ patches) and a 10 class subset of ImageNet (Figure 5b, using $3\times7\times7$ patches). The label vector is 1 for a given class and zero for all other classes. Correlations are all around 0, suggesting the assumption that patches are uncorrelated with their labels is true for real datasets.

5. Comparing Theory to Practice

The theory in the previous section used a highly simplified CNN trained with MSE loss. We now ask: how well does the theory predict the energy profiles of real-world, nonlinear CNNs trained with the standard cross-entropy loss?

A major assumption in our theory was that the labels are uncorrelated with individual PCA coefficients. This is obviously true for random labels, but we wanted to check whether it was also true for true labels in commonly used datasets. Figure 5 measures this correlation in CIFAR10 and in a 10 class subset of ImageNet. Specifically we consider 10 "one vs. all" binary classification tasks. For each

such task, we measure the correlation between the label and each of the individual PCA coefficients of patches in the image. For CIFAR10 we use $3 \times 3 \times 3$ patches (or 27 PCA components) and for ImageNet we use $7 \times 7 \times 3$ patches (or 147 components). The figure plots these correlation coefficients for different binary classification tasks and for all PCA coefficients. For all classes, correlation with the labels is close to 0 for all components, supporting our assumption.

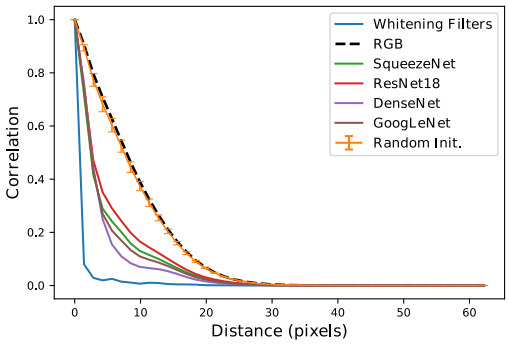
5.1. Spatial Decorrelation in CNNs

One prediction from our analysis is that the first layer of CNNs should perform partial decorrelation. Let $y_i(x)$ be the vector that denotes the output of all channels at a particular location i for image x. The autocorrelation function is defined as $C(\delta) = \mathbb{E}_{i,x}[y_i(x)y_{i+\delta}(x)]$ where the expectation is taken over locations (i) and training images. It is easy to show that if y is obtained from x by a whitening transform, then the autocorrelation function should be a 0 for any $\delta \neq 0$. We wanted to see if this holds for real-world CNNs.

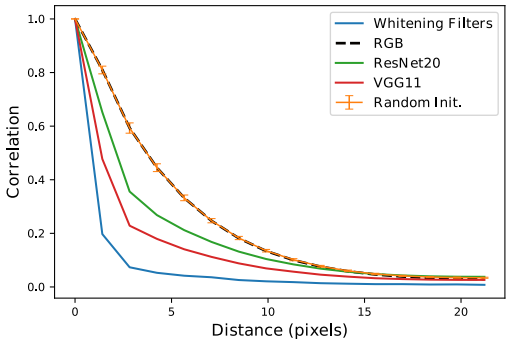
We first compute this autocorrelation when y includes three channels corresponding to the input (R,G,B). As can be seen in Figure 6 the correlation decreases as δ increases, but even at a distance of 10 pixels the correlation is above 0.5. When we measure this same autocorrelation function with 64 random filters (i.e. the first layer of commonly used CNNs at initialization), the vector y is of length 64, but the autocorrelation function is almost identical to that of RGB (note that the graph corresponding to random weights includes error bars and summarizes 100 different random initializations but all random initializations give very similar autocorrelation functions). In contrast, when the vector y is the output of all 64 channels in the first layer of a *learned* CNN, we consistently find that the spatial correlation is significantly reduced, (e.g. at a distance of 10 pixels the correlation after learning is reduced to around 0.2). For comparison, we also show the autocorrelation function of a set of filters that satisfy perfect whitening which reduces the correlation at distance 10 pixels to zero, as expected. Thus, consistent with our theoretical analysis of linear CNNs, real-world CNNs perform approximate whitening of the input and remove much of the redundancy that is present in their input even though they are not explicitly trained with a redundancy reduction loss.

5.2. Fitting the Formula to CNNs

Not only does our analysis predict this partial decorrelation of the input at a finite number of training iterations, it also gives a precise characterization of the energy profiles for a linear CNN. Does this formula predict the energy profiles of real models? We compare Equation (4) to energy profiles of real models by setting a constant learning rate for all datasets



(a) ImageNet

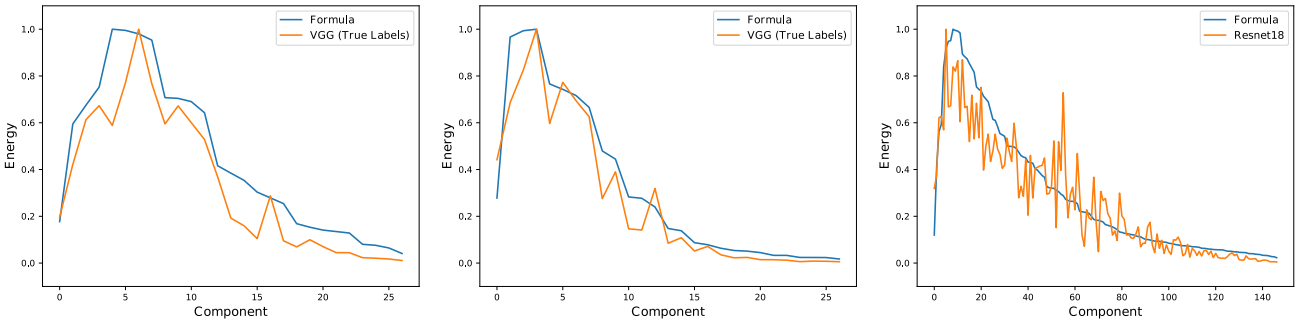


(b) CIFAR10

Figure 6. Auto-correlation as a function of distance for different representations of the input. In the RGB representation and in a first layer that has random weights, the autocorrelation is significant at large distances, but as the network is trained, this spatial redundancy is reduced. Thus the first layer learns to perform partial "redundancy reduction" as predicted by our analysis.

and searching over the number of gradient steps t. The results, portrayed in Figure 7 show *high correlation between the formula and real-world models* (consistently above 0.9), even in complex datasets such as ImageNet. Section 5.2 expands on these by providing correlation coefficients of the formula to different models, with multiple random seeds and on many datasets. Consistently, the formula calculated at a finite iteration is able to capture much of what is done by the first layer, independent of dataset, but not that of a random initialization. More fits for ImageNet, CIFAR10, CIFAR100, MNIST and for unsupervised tasks are provided in Appendix B.

Additionally to capturing the profile of the first layer of trained models, our formula also captures the *dynamics* of gradient descent. Figure 8 shows an excellent fit between the formula at different iterations of gradient descent and the profile of the first layer of VGG11 trained on CIFAR10 and MNIST during training. Clearly, as the training of the model progresses, so does the number of iterations required for the formula to fit its profile, showing a correspondence between



(a) VGG11 on CIFAR10 (0.96) (b) VGG11 on CelebA (0.97) (c) ResNet18 on Imagenet (0.92)

Figure 7. Energy profiles of deep, nonlinear CNNs (orange) and the energy profile predicted by Equation (4) (blue). Even though the formula was derived for a simple, linear CNN, the correlation coefficient between the predicted and observed profiles is often above 0.9. See Section 5.2 for more correlations.

the two training dynamics. Furthermore, as the number of iterations increases, the energy profile of real-world CNNs approaches a *whitening* profile of the first components.

Our theory predicts that if gradient descent is run for an infinite number of epochs, the learned weights will eventually converge to whitening of all components. In our experiments, we only observed partial whitening even after 10,000 training epochs and we believe this is due to the fact that the gradient of loss with respect to the weights in the first layer becomes extremely small after a finite number of epochs (ratio of $\sim 10^{-10}$) causing the dynamics to plateau.

DATASET	CORRELATION
IMAGENET	0.9±0.01
CIFAR10	0.94±0.01
CELEBA	0.96±0.01
CAR VS TRUCK	0.93±0.01
DOG VS FROG	0.91±0.02
DOG VS CAT	0.95±0.01
BIRD VS PLANE	0.96±0.005
BOAT VS PLANE	0.96±0.01
RANDOM INIT.	0.1±0.15

Table 3. Correlation between energy profiles of VGG11 with the analytic formula for CIFAR10, CelebA and different binary datasets of CIFAR10, averaged over 3 different seeds. Correlations for ImageNet are averaged over 5 different models (and see Appendix B). The correlation with a random initialization is also presented for reference.

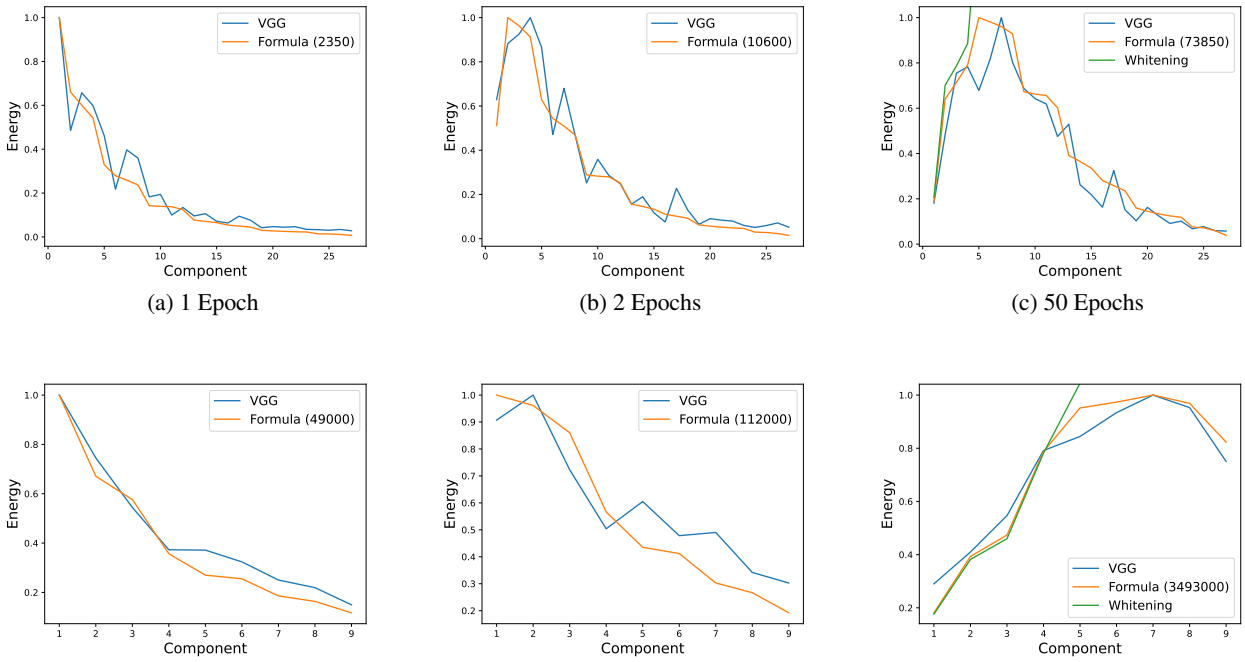
5.3. Changing the Data Statistics

In a final test of the ability of our analytic formula to fit the energy profiles of real, nonlinear CNNs we design two experiments that attempt to change the energy and label statistics of the classification task. In the first task we force the true labels to correlate with the input - for the i'th PCA component, we sort all CIFAR10 images by their average energy in the i'th direction and divide them into 10 equally sized sets. This creates 27 datasets (as the number of PCA components for $3 \times 3 \times 3$ patches), each with high correlation between the labels and the image energy. Figure 9a displays the result of this experiment conducted on the 15'th component. As expected, the profiles of true and random labels are now noticeably different and their correlation drops to around 0 (and see Appendix C.1), only by introducing correlation between the images' patch energy and labels. Additionally, the first layer changes to be extremely sensitive to the specific component that is correlated with the labels.

In another experiment, we change only the input statistics by multiplying each patch by a constant factor α in a specific PCA direction. In this setting, there is no change in correlation between the patch energy and the labels as the same transformation is applied to all patches, and the eigenvalue corresponding to the component we enhanced is changed from λ_i^2 to $\alpha^2\lambda_i^2$. Figure 9b shows that as expected, our analytic formula for random labels still captures the energy profiles of VGG with true labels, after applying the same transformation that was done to the input images to the eigenvalues used in the formula. More results are presented in Appendix C.2.

6. Related Works

There have been many studies devoted to comparing representations in different neural networks (Laakso & Cottrell, 2000; Lenc & Vedaldi, 2015; Csiszárik et al., 2021). The comparison is often done by comparing the output of transformations induced by the neurons (Kornblith et al., 2019; Nguyen et al., 2021; Doimo et al., 2020) or the neurons themselves (Wang et al., 2018; Li et al., 2015). The energy profile is an alternative method that is especially useful for



(a) 1 Epoch (b) 2 Epochs (c) 50 Epochs

(d) 10 Epochs (e) 25 Epochs (f) 200 Epochs

Figure 8. Energy profiles of VGG11 (blue) compared to the profile predicted by Equation (4) (orange) at different number of epochs. Both for training on CIFAR10 with true labels (top) and for training with MNIST and random labels (bottom) the profiles are highly consistent and approach whitening on the first components. All correlations between the profiles and the formula are above 0.95. Initialization was subtracted from the model to simulate zero initialization.

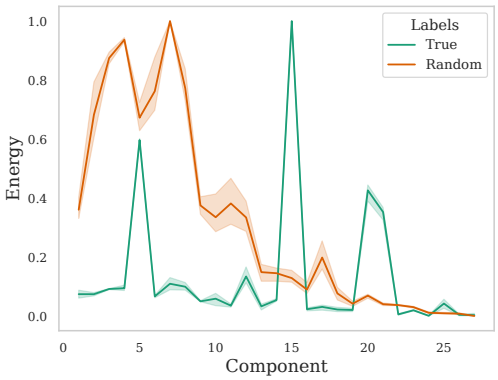
comparing linear representations and avoids many of the pitfalls of previous approaches.

The fact that different CNNs tend to learn qualitatively similar filters in the first layer has been reported previously (Yosinski et al., 2014; Sarwar et al., 2017; Luan et al., 2017; Alekseev & Bobe, 2019, for example), and follows from a line of work of visualizing representations in deep CNNs (Zeiler & Fergus, 2013; Girshick et al., 2013). Our work extends this finding by showing that the overall representation in the first layer is not only qualitatively but also is *quantitatively* similar - different CNNs not only learn to recognize spatial frequencies in their first layer but also the same distribution of frequencies. This consistency is then expanded to networks trained with true and random labels.

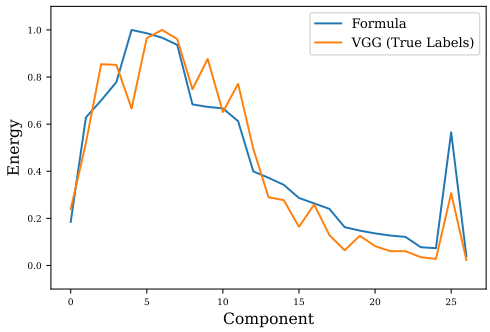
The idea that early representations should remove redundancies in their input goes back to Barlow (1989) and there has been a great deal of work arguing that initial layers in biological neural networks remove dependencies in their input (Field, 1994; Olshausen & Field, 1996; Bell & Sejnowski, 1997). In particular, when explicit redundancy reduction is performed on natural image data, this principle leads to Gabor filters similar to those that are observed in the first layer of CNNs. In this work we followed Atick & Redlich (1990) and focused on removing linear dependencies by whitening. More importantly, we have shown that this form of redundancy reduction emerges from minimizing the classification loss either with true or random labels.

The usefulness of whitening as a normalization step in image processing techniques is well known (Hyvärinen et al., 2009), and is even used as a preprocessing technique when training CNNs (Coates et al., 2011; Pal & Sudeep, 2016). This has inspired others to constrain intermediate representations of neural networks to be white as well (Desjardins et al., 2015; Luo, 2017; Huang et al., 2018; Pan et al., 2019; Zhang et al., 2021) in order to improve convergence time and performance. Our work shows that approximate whitening occurs in CNNs even without an explicit whitening preprocessing step nor without an explicit "redundancy reduction" loss.

As previously explained, the emergence of whitening is partially the result of a bias in the gradient descent training algorithm. The fact that gradient descent training biases towards certain solutions has been known for many years, and proven mainly for linear predictors and separable data. Studies on linear networks (Soudry et al., 2018) and linear CNNs (Gunasekar et al., 2018) found that under certain con-



(a) Labeling by the 15'th Component



(b) Enhancing the 25'th Component

Figure 9. Changing the joint distribution of the patch energy and the labels affects the energy profiles. When correlation between the energy and the labels is introduced (9a) then the correlation between the true and random profiles is broken. When statistics are changed without introducing correlation (9b) the theory follows suit. More results can be found in Appendix C

ditions, gradient descent causes the effective linear predictor to be biased towards sparsity (in Fourier space in the case of CNNs) or minimal norm or max-margin (Chizat & Bach, 2020). Similar works have also shown that deep nonlinear networks are biased towards learning lower frequencies first (Rahaman et al., 2019). Our theoretical analysis follows this line, and that of gaining insight into real-world networks from simpler linear models (LeCun et al., 1991; Hacohen & Weinshall, 2022; Gidel et al., 2019; Gissin et al., 2019), while verifying our claims by quantitatively showing consistency between theory and practice.

Previous works have examined the usefulness of representations in models trained with random labels by incorporating them in transfer learning. Indeed, we show explicitly that since there is a high degree of similarity between the first layer of models trained with true labels and random ones, it is reasonable to assume that layers of random models could be useful for transfer learning. While some claimed (Bansal et al., 2021) that this was due to similarity between the first layer of a model trained with random labels and a random initialization, Maennel et al. (2020) offered the explanation that the first layer filters' covariance and the patch PCA have the same eigenvectors. Our results contradict the hypothesis of (Bansal et al., 2021) and extend the results of (Maennel et al., 2020) to give an analytic formula for the energy profile that holds for true and random labels.

7. Discussion

The dramatic success of CNNs has led to increased interest in the representations they learn, whether for explainability or for transferring between different tasks. In this paper we have focused on the representation that CNNs learn in the very first layer and presented a high degree of quantitative consistency between the energy profiles learned by different networks using different initializations, architectures, and even labels. To understand why CNNs learn this particular energy profile we analyzed linear CNNs and showed that this consistency is not a result of usefulness for object recognition but rather due to properties of the input and output statistics. Specifically the profile is mostly due to the lack of correlation between image patches and labels and the bias of the training algorithm. Combined, the two give an implicit bias towards partial "redundancy reduction".

To generalize to real-world CNNs, we showed that the analytic formulation of the linear case captures much of what is done by the first layer of different networks on different datasets. To complement our explanation, we designed experiments that adjust the statistics of the input and output and showed the results behave as predicted.

Redundancy reduction is closely related to what is commonly referred to as "disentanglement" in deep learning (Goodfellow et al., 2016): representations of the input should disentangle the different factors of variation that influence each piece of the input. Our results show that real-world CNNs trained with gradient descent perform a simplified version of disentanglement even if there is no explicit loss that rewards it. It will be interesting to see if this result can be extended to deeper layers and more nonlinear definitions of disentanglement.

Acknowledgements

We gratefully appreciate the Gatsby Foundation for their support in funding this research. We also thank Roy Friedman for his helpful insights.

References

Alekseev, A. and Bobe, A. Gabornet: Gabor filters with learnable parameters in deep convolutional neural network. In *2019 International Conference on Engineering and Telecommunication (EnT)*, pp. 1–4, 2019. doi: 10.1109/EnT47717.2019.9030571.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., and Lacoste-Julien, S. A closer look at memorization in deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 233–242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/arpit17a.html.

Atick, J. J. and Redlich, A. N. Towards a Theory of Early Visual Processing. *Neural Computation*, 2(3):308–320, 09 1990. ISSN 0899-7667. doi: 10.1162/neco.1990.2.3.308. URL https://doi.org/10.1162/neco.1990.2.3.308.

Bansal, Y., Nakkiran, P., and Barak, B. Revisiting model stitching to compare neural representations. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 225–236. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/01ded4259d101feb739b06c399e9cd9c-Paper.pdf.

Barlow, H. Unsupervised Learning. *Neural Computation*, 1(3):295–311, 09 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.3.295. URL https://doi.org/10.1162/neco.1989.1.3.295.

Bell, A. J. and Sejnowski, T. J. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997. ISSN 0042-6989. doi: https://doi.org/10.1016/S0042-6989(97)00121-1. URL https://www.sciencedirect.com/science/article/pii/S0042698997001211.

Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Abernethy, J. and Agarwal, S. (eds.), *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 1305–1338. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/chizat20a.html.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Gordon, G., Dunson, D., and Dudík, M. (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

Csiszárik, A., Kőrösi-Szabó, P., Matszangosz, Á. K., Papp, G., and Varga, D. Similarity and matching of neural network representations. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=aedFIIRRfXr.

Desjardins, G., Simonyan, K., Pascanu, R., and kavukcuoglu, k. Natural neural networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Ding, F., Denain, J.-S., and Steinhardt, J. Grounding representation similarity through statistical testing. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=_kwj6V53ZqB.

Doimo, D., Glielmo, A., Ansuini, A., and Laio, A. Hierarchical nucleation in deep neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7526–7536. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/54f3bc04830d762a3b56a789b6ff62df-Paper.pdf.

Field, D. J. What is the goal of sensory coding? *Neural computation*, 6(4):559–601, 1994.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018a. URL http://arxiv.org/abs/1803.07728.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018b. URL https://openreview.net/forum?id=S1v4N2l0-.

Gidel, G., Bach, F., and Lacoste-Julien, S. Implicit regularization of discrete gradient dynamics in linear neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f39ae9ff3a81f499230c4126e01f421b-Paper.pdf.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013. URL http://arxiv.org/abs/1311.2524.

Gissin, D., Shalev-Shwartz, S., and Daniely, A. The implicit bias of depth: How incremental learning drives generalization. *CoRR*, abs/1909.12051, 2019. URL http://arxiv.org/abs/1909.12051.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/0e98aeeb54acf612b9eb4e48a269814c-Paper.pdf.

Hacohen, G. and Weinshall, D. Principal components bias in over-parameterized linear models, and its manifestation in deep neural networks. *Journal of Machine Learning Research*, 23(155):1–46, 2022. URL http://jmlr.org/papers/v23/21-0991.html.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Huang, L., Yang, D., Lang, B., and Deng, J. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Hyvärinen, A., Hurri, J., and Hoyer, P. O. (eds.). *Natural Image Statistics*. Springer-Verlag London, London, UK, 2009.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Laakso, A. and Cottrell, G. Content and cluster analysis: Assessing representational similarity in neural systems. *Philosophical Psychology*, 13(1):47–76, 2000. doi: 10.1080/09515080050002726. URL https://doi.org/10.1080/09515080050002726.

LeCun, Y., Kanter, I., and Solla, S. Second order properties of error surfaces: Learning time and generalization. In Lippmann, R. P., Moody, J., and Touretzky, D. (eds.), *Advances in Neural Information Processing Systems*, volume 3. Morgan-Kaufmann, 1991. URL https://proceedings.neurips.cc/paper/1990/file/758874998f5bd0c393da094e1967a72b-Paper.pdf.

Lenc, K. and Vedaldi, A. Understanding image representations by measuring their equivariance and equivalence. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 991–999, Los Alamitos, CA, USA, jun 2015. IEEE Computer Society. doi: 10.1109/CVPR.2015.7298701. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298701.

Li, Y., Yosinski, J., Clune, J., Lipson, H., and Hopcroft, J. Convergent learning: Do different neural networks learn the same representations? In Storcheus, D., Rostamizadeh, A., and Kumar, S. (eds.), *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pp. 196–212, Montreal, Canada, 11 Dec 2015. PMLR. URL https://proceedings.mlr.press/v44/li15convergent.html.

Luan, S., Zhang, B., Chen, C., Cao, X., Han, J., and Liu, J. Gabor convolutional networks. *CoRR*, abs/1705.01450, 2017. URL http://arxiv.org/abs/1705.01450.

Luo, P. Learning deep architectures via generalized whitened neural networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2238–2246. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/luo17a.html.

Maennel, H., Alabdulmohsin, I. M., Tolstikhin, I. O., Baldock, R., Bousquet, O., Gelly, S., and Keysers, D. What do neural networks learn when trained with random labels? In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19693–19704. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e4191d610537305de1d294adb121b513-Paper.pdf.

Nguyen, T., Raghu, M., and Kornblith, S. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Olshausen, B. A. and Field, D. J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

Owsley, C. Contrast sensitivity. *Ophthalmology clinics of North America*, 16(2):171—177, June 2003. ISSN 0896-1549. doi: 10.1016/s0896-1549(03)00003-8. URL https://doi.org/10.1016/s0896-1549(03)00003-8.

Pal, K. K. and Sudeep, K. S. Preprocessing for image classification by convolutional neural networks. In *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1778–1781, 2016. doi: 10.1109/RTEICT.2016.7808140.

Pan, X., Zhan, X., Shi, J., Tang, X., and Luo, P. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Rahaman, Baratin, Arpit, Draxler, Lin, a. H., Bengio, and Courville. On the spectral bias of neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html.

Sarwar, S. S., Panda, P., and Roy, K. Gabor filter assisted energy efficient fast learning convolutional neural networks. *CoRR*, abs/1705.04748, 2017. URL http://arxiv.org/abs/1705.04748.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Soudry, D., Hoffer, E., and Srebro, N. The implicit bias of gradient descent on separable data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1q7n9gAb.

Wang, L., Hu, L., Gu, J., Hu, Z., Wu, Y., He, K., and Hopcroft, J. Towards understanding learning representations: To what extent do different neural networks learn the same representation. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/5fc34ed307aac159a30d81181c99847e-Paper.pdf.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/375c71349b295fbe2dcdca9206f20a06-Paper.pdf.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. URL http://arxiv.org/abs/1311.2901.

Zhang, S., Nezhadarya, E., Fashandi, H., Liu, J., Graham, D., and Shah, M. Stochastic whitening batch normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10978–10987, June 2021.

A. Proofs of Theorems on CNNs

We start with some definitions. To simplify the notation, we assume that the mean of the patches in the training set is zero.

Definition A.1. Given a set of patches $\{p_n\}$ the PCA vectors u_i are eigenvectors of the matrix $\sum_n p_n p_n^T$.

Definition A.2. Given a set of filters $\{w_k\}$ and a set of PCA vectors $\{u_i\}$ the energy profile of the set is given by a vector e whose ith component is given by:

$$e_i^2 = \frac{1}{K}\sum_{k=1}^{K}(w_k^T u_i)^2 \tag{5}$$

Definition A.3. Given a set of patches $\{p_n\}$ and a set of PCA vectors $\{u_i\}$ the energy profile of the set is given by a vector λ whose ith component is given by:

$$\lambda_i^2 = \frac{1}{N}\sum_{n=1}^{N}(p_n^T u_i)^2 \tag{6}$$

Definition A.4. A labeled training set of images $\{x_n, y_n\}$ satisfies the property that the label is uncorrelated with individual PCA coefficients if $E\left[u_i^T p_j(x) y(x)\right] = E\left[u_i^T p_j(x)\right] E\left[y(x)\right]$ where the expectation is over the dataset and $p_j(x)$ is a randomly chosen patch in image x.

Theorem A.5. *Consider a depth-2 linear CNN of any width initialized with zero mean filters and variance $\sigma^2 I$ and trained with gradient descent with step size η on the MSE loss. Assume that different patches in each image are uncorrelated with each other and that the labels are uncorrelated with individual PCA components, then as the number of patches in the training set goes to infinity, the energy profile of the filters at iterations t is given by:*

$$e_i = \tilde{c}\cdot\frac{|1-(1-\eta\lambda_i^2)^t|}{\eta^2\lambda_i^2}\lambda_i + \xi_i \tag{7}$$

where λ_i is the energy profile of the training patches and ξ a random vector that depends on the initialization and whose magnitude goes to zero as $\sigma \to 0$.

Proof. The output of the network for an input image x is given by:

$$\hat{y}(x) = \sum_k \frac{1}{J}\sum_{j=1}^{J} p_j(x)^T w_k = c\bar{p}^T(x)\bar{w} \tag{8}$$

where $p_j(x)$ is the jth patch in image x, $\bar{p}(x)$ is the average patch in image x and $\bar{w}$ is the average filter and c is the number of filters. This also means that the gradient of the MSE loss $L = \frac{1}{N}\sum_x (y(x)-\hat{y}(x))^2$ with respect to a particular filter is given by:

$$\frac{\partial L}{\partial w_k} = c\left(A\bar{w} - b\right) \tag{9}$$

where $A = \frac{1}{N}\sum_x \bar{p}(x)\bar{p}(x)^T$ and $b = \frac{1}{N}\sum_x \bar{p}(x)y(x)$. Note that the gradient is the same for all k which means that at each iteration:

$$w_k(t) = \bar{w}(t) + w_k(0) \tag{10}$$

and we can describe the dynamics of the mean filter at each iteration t by:

$$\bar{w}(t) = \bar{w}(t-1) - \eta\left(A\bar{w}(t-1) - b\right) \tag{11}$$

Defining the matrix $C = (I - \eta A)$ and assuming that the mean filter at the initial iteration is 0 gives:

$$\bar{w}(t) = \left(\sum_{n=0}^{t-1} C^n\right) b \tag{12}$$

Note that the matrix C is diagonalized by the PCA basis and its eigenvalues are $1-\eta\lambda_i$ which means that:

$$u_i^T\bar{w}(t) = \frac{\left(1-(1-\eta\lambda_i^2)^t\right)}{\eta^2\lambda_i^2}(u_i^T b) \tag{13}$$

Or taking the absolute value of both sides:

$$|u_i^T\bar{w}(t)| = |\frac{\left(1-(1-\eta\lambda_i^2)^t\right)}{\eta^2\lambda_i^2}|\cdot|(u_i^T b)| \tag{14}$$

Now consider the term $|u_i^T b|$ this can be rewritten:

$$|u_i^T b| = \left|\frac{1}{N}\sum_x u_i^T\bar{p}(x)y(x)\right| \tag{15}$$

By the central limit theorem, the term $z_i = \frac{1}{N}\sum_x u_i^T\bar{p}(x)y(x)$ approaches a Gaussian whose mean is the mean of the random variable $y(u_i^T p)$, i.e. the random variable is the product of the label of an image and a PCA coefficient of the average patch in that image. Since we are assuming the labels to be uncorrelated with the PCA coefficient, the mean of this random variable is 0 and its variance is λ_i^2/J (where J is the number of patches). Thus z_i is a Gaussian random variable with mean zero and variance $\lambda_i^2/(JN)$ and the term $|u_i^T b|$ is a "folded Gaussian" whose expectation is:

$$\mathbb{E}\left[\left|u_i^T b\right|\right] = \frac{\lambda_i}{\sqrt{JN}}\frac{\sqrt{2}}{\sqrt{\pi}} \tag{16}$$

and whose variance is also proportional to $1/JN$. As $JN\to\infty$, the variance goes to zero which means that $|u_i^T b|$ is with high probability close to its expected value and hence $|u_i^T b|$ is with high probability proportional to λ_i.

Substituting this in equation 17 gives that with high probability:

$$\left|u_i^T\bar{w}(t)\right| = c_2\left|\frac{\left(1-(1-\eta\lambda_i^2)^t\right)}{\eta^2\lambda_i^2}\right|\lambda_i \tag{17}$$

Finally, by the definition of the energy profile and the fact that $w_k(t) = \bar{w}(t) + w_k(0)$ equation 7 follows. □

Theorem A.6. *Let $\{w_k\}$ be the filters in the first layer of a CNN. If the energy profile of these filters satisfy Equation (7) then as the number of iterations goes to infinity, the filters in the first layer of the CNN perform spatial decorrelation.*

Proof. It is evident from Equation (7) that as $t\to\infty$, the energy profile is proportional to $\frac{1}{\lambda}$. This means that the filter bank performs "whitening" and there have been many works that show the connection of whitening to spatial decorrelation (see Hyvärinen et al. (2009) and references within). For completeness, we give the derivation here.

Recall that the PCA vectors of natural image patches aare approximately the Fourier basis. Thus the fact that the energy profile is proportional to $\frac{1}{\lambda}$ implies a relationship between the Fourier transform of the bank of filters and the Fourier transform of the images. Denote by $\mathbb{E}\left[\left|x^F(\omega)\right|\right]$ the expected power spectrum of the training images and by $\left|w_k^F(\omega)\right|$ the power spectrum of the k'th filter then:

$$\sum_k\left|w_k^F(\omega)\right|^2 \propto \frac{1}{\mathbb{E}\left[\left|x^F(\omega)\right|^2\right]} \tag{18}$$

Now denote by C the auto-correlation function of the representation and by y_k the k'th channel activations, i.e. $y_k = x\star w_k$ then:

$$C = \mathbb{E}_x\left[\sum_k y_k\star y_k\right] \tag{19}$$

where the expectation is over images in the training set. We say that a representation is "spatially disentangled" if the channels at different locations are uncorrelated and C is a delta function.

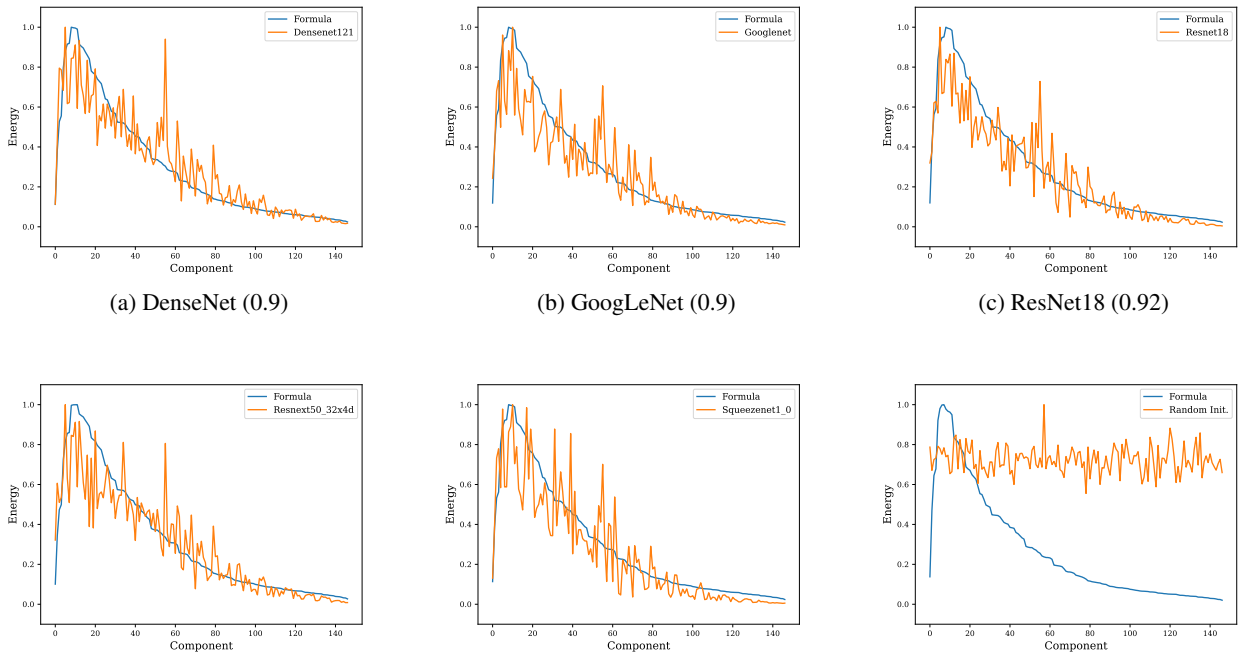
We denote by $C^F(\omega)$ the Fourier Transform of C and $y_k^F(\omega)$ are the Fourier transforms of each channel. Then:

$$
\begin{aligned}
C^F(\omega) &= \mathbb{E}\left[\sum_k \left|y_k^F(\omega)\right|^2\right] && (20)\\
&= \sum_k \left|w_k^F(\omega)\right|^2 \mathbb{E}\left[\left|x^F(\omega)\right|^2\right] && (21)\\
&= \mathbb{E}\left[\left|x^F(\omega)\right|^2\right] \sum_k \left|w_k^F(\omega)\right|^2 && (22)\\
&= c && (23)
\end{aligned}
$$

Where the last equation is derived by substituting Equation (18). Hence the Fourier Transform of the auto-correlation function is a constant which means that the auto-correlation function is a δ function. □

B. Fitting Formula to Different Models

To expand on the results in Section 5, presented are more fits of the formula in Equation (7) to different models on different datasets. Figure 10 depicts models trained on ImageNet, which have been downloaded from the PyTorch model hub, which are highly correlated with the theoretical formula. Meanwhile, a random initialization can hardly be explained using it. Figure 11 and Figure 13 provide more examples of fitting the formula to models trained on CIFAR10 and CIFAR100 respectively. Figure 12 shows the formula fitted to pretrained models on CIFAR10, and see Appendix E for more information. Figure 14 shows fits of the formula to a model trained on MNIST over different iterations. An additional fit to a self-supervised model is presented in Figure 15.



(a) DenseNet (0.9) (b) GoogLeNet (0.9) (c) ResNet18 (0.92)

(d) ResNeXt (0.9) (e) SqueezeNet (0.9) (f) Random Init. (0.11)

Figure 10. Fitting Equation (7) to different models trained on ImageNet by searching over iterations. An example of a random initialization is attached for reference. Correlation coefficients in parentheses.

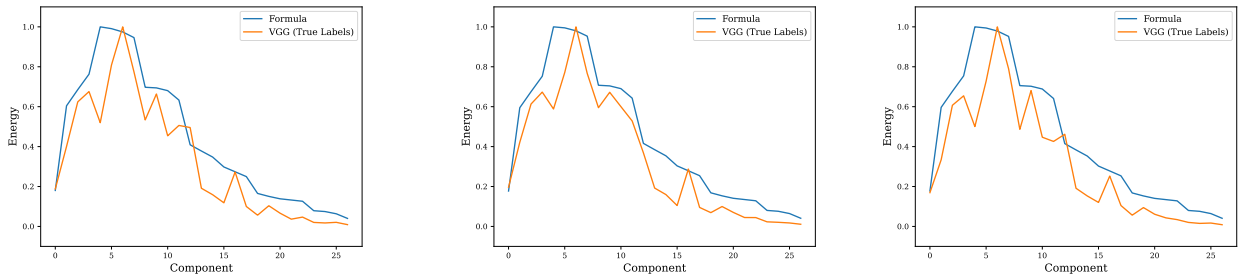
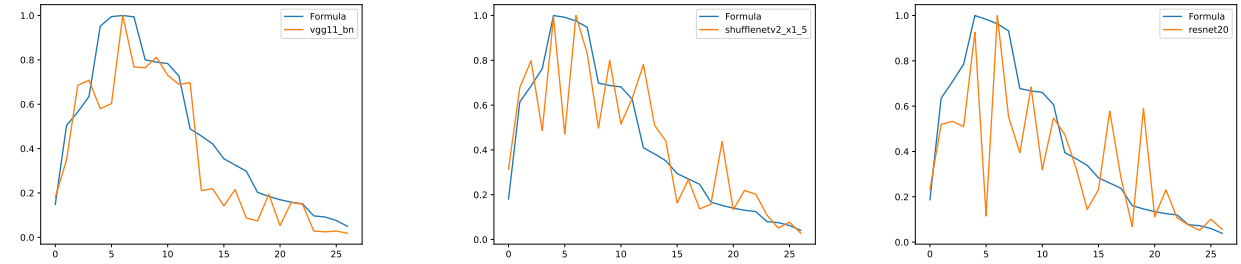


Figure 11. More examples of fitting Equation (7) to VGG11 trained on CIFAR10 with different random seeds. Correlations are above 0.94.



(a) VGG11 (0.91) (b) ShuffleNet (0.85) (c) ResNet20 (0.69)

Figure 12. Fitting the formula to pretrained models trained on CIFAR10 (and see Appendix E for more). These models were trained with learning rate schedulers, weight decay and momentum, all of which not covered in our theory and can cause differences in practice.

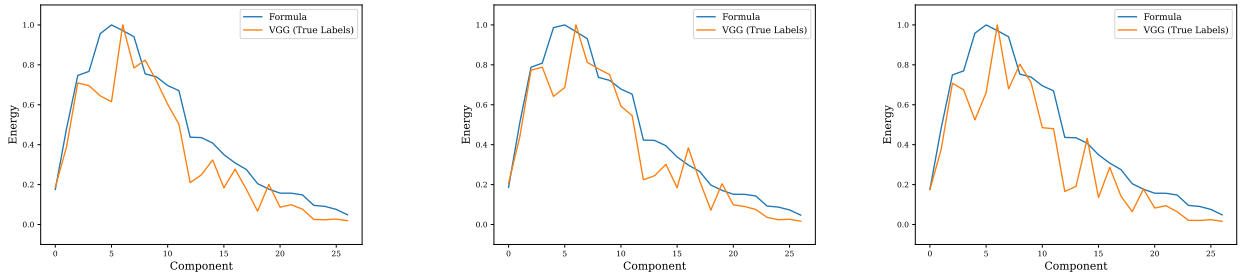
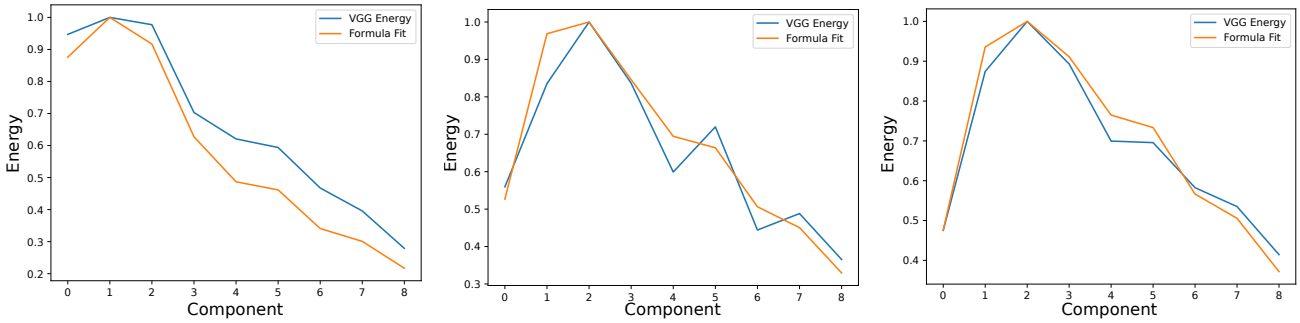


Figure 13. Examples of fitting Equation (7) to VGG11 trained on CIFAR100 with different random seeds. Correlations are above 0.93.



(a) Epoch 1 (0.99) (b) Epoch 3 (0.96) (c) Epoch 25 (0.99)

Figure 14. Fitting Equation (7) to VGG trained on MNIST with true labels, at different iterations. Correlations are in parentheses.

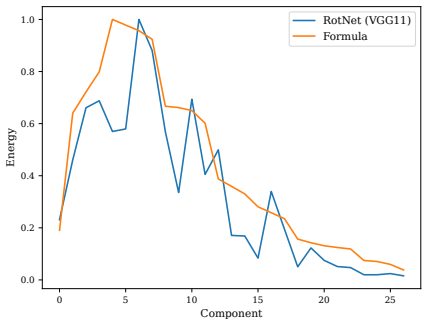


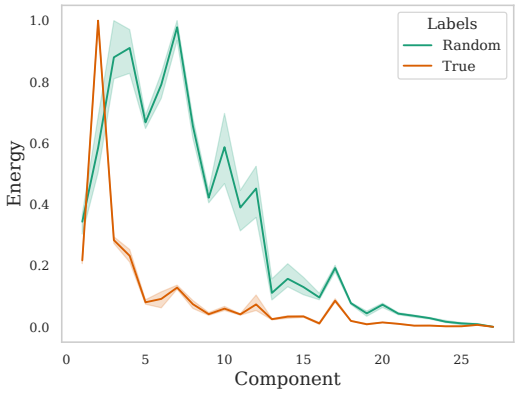
Figure 15. Fitting Equation (7) to "RotNet" (Gidaris et al., 2018b) - VGG11 trained to predict image rotations on CIFAR10.

C. Effects of Changing Label and Image Statistics

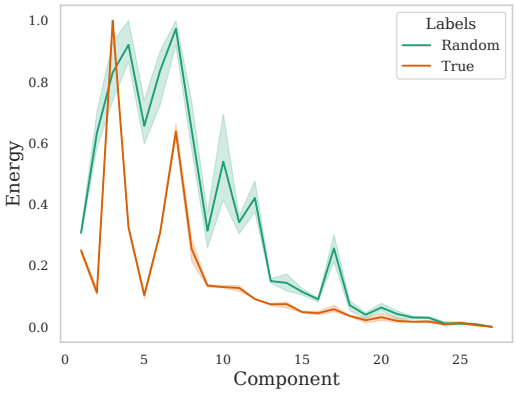
As explained in Section 5, we conducted two experiments changing the input-output statistics and testing the effects on the learned energy profiles. According to Theorem 4.2, as long as the PCA components remain uncorrelated we expect models trained with true and random labels to remain consistent with each other and with the formula in Equation (7).

C.1. Introducing Correlation between Patches and Labels

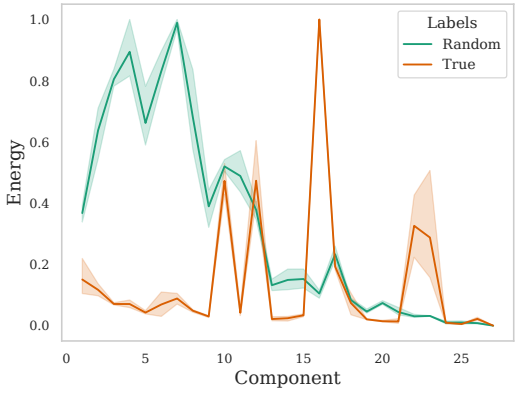
In the first, each image was labeled according to the energy w.r.t. a PCA component u. For an image X with patches $P_1(X)...P_k(X)$ we calculated the quantity $\sum_{i=1}^{k}(P_i(X)^T u)^2$ to be the total patch energy in direction u, and labeled X according to the percentile of its energy (top 10% of images w.r.t. their energy were labeled $y = 1$, bottom 10% were labeled $y = 10$ and so on).
Figure 16 shows that for different components, the correlation drops between the profiles of models trained with the new true labels and random labels. Notice, the decrease is more is larger when the labels are determined by components which aren't learned by the model with random labels.



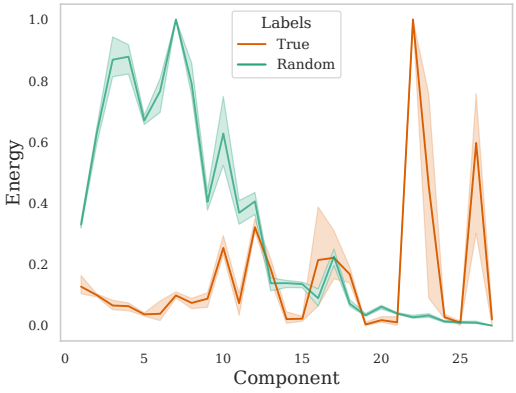
(a) Component 2 (0.44)



(b) Component 3 (0.76)



(c) Component 16 (-0.06)



(d) Component 26 (-0.24)

Figure 16. Training with true and random labels, when the true labels correspond to the image patch energy in different components (mean correlation in parenthesis). Once introducing correlation between patches and labels, profiles of true and random labels cease to correlate.

C.2. Changing the Patch Statistics

In this experiment, we changed the patch distribution consistently for all classes, therefore not changing the correlation between energy and labels. Let $u_1 ... u_d$ be the PCA components. Therefore each patch $P_i(X)$ of and image X can be spanned as:

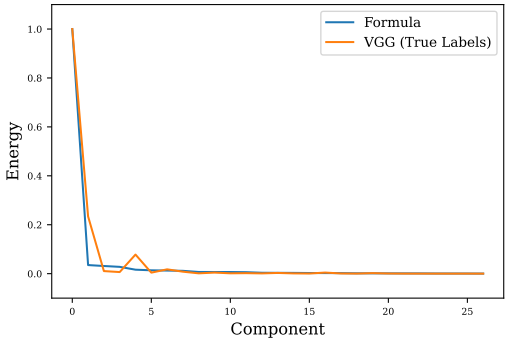
$$P_i(X) = \sum_{j=1}^{d} \langle P_i(X), u_j \rangle \, u_j \tag{24}$$

We adjust the distribution by constant $\alpha > 1$ w.r.t. component u_t by transforming:

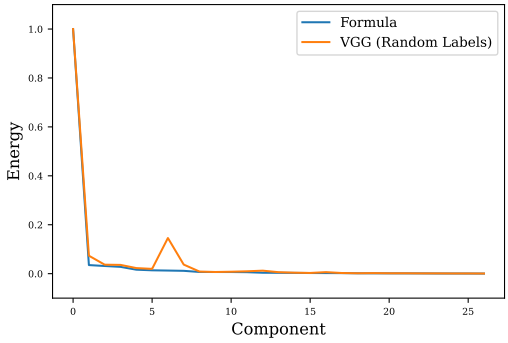
$$\sum_{j=1}^{d} \langle P_i(X), u_j \rangle \, u_j \rightarrow \alpha \, \langle P_i(X), u_t \rangle \, u_t + \sum_{j \neq t} \langle P_i(X), u_j \rangle \, u_j \tag{25}$$

Therefore changing the patch PCA eigenvalue corresponding to u_t. We do this to all overlapping patches in the dataset (therefore with no affect to the correlation between patches and labels), and adjust the stride in the first layer to avoid issues with the overlap.

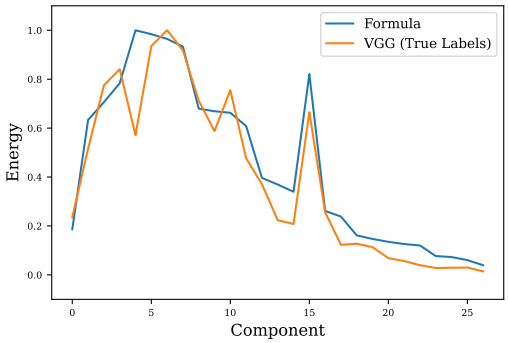
Figure 17 shows that indeed the first layers of models trained with true and random labels are still highly similar after applying the transformation in Equation (25). Profiles of both models can still be explained by our analytic formula after applying the same transformation to the eigenvalues used to calculate it.



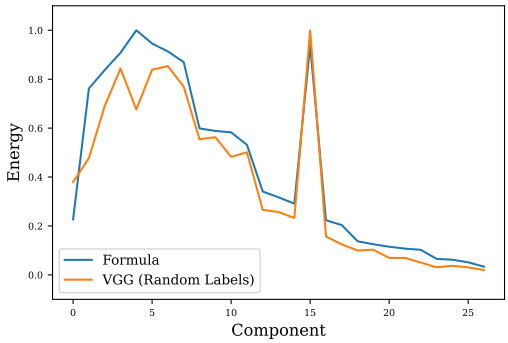
(a) Component 0 (True)



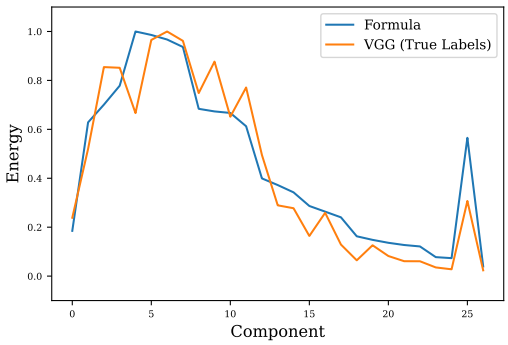
(b) Component 0 (Random)



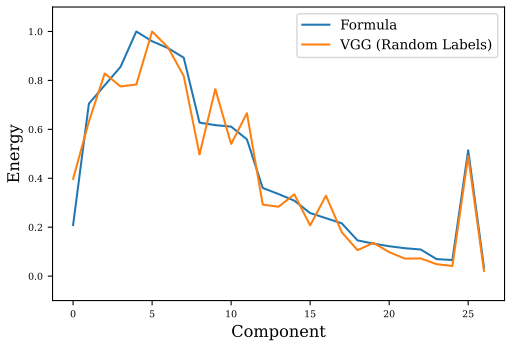
(c) Component 15 (True)



(d) Component 15 (Random)



(e) Component 25 (True)

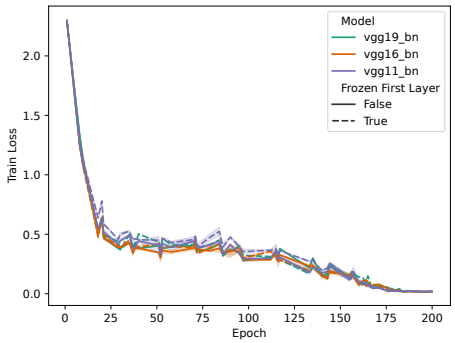


(f) Component 25 (Random)

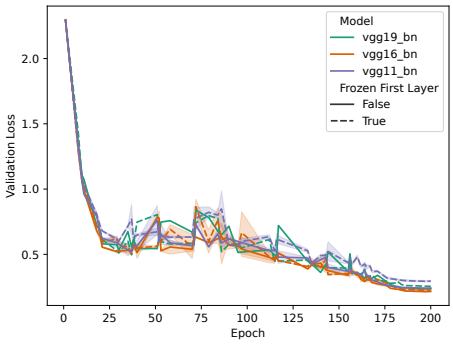
Figure 17. Changing the eigenvalues corresponding to different PCA component for true and random labels. The profiles for both sets of labels are highly similar and can be explained by the analytic formula.

D. CNNs with Frozen First Layer

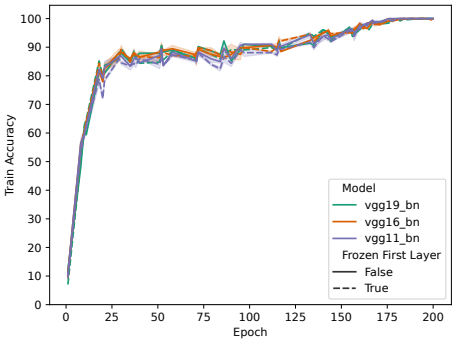
To expand on the result on VGGs with a frozen first layer we attach here the full set of training results. As can be seen in Figure 18, as the depth of a network increases the difference between the model with a frozen first layer and a learnt one is almost indistinguishable - both in terms of accuracy and loss.



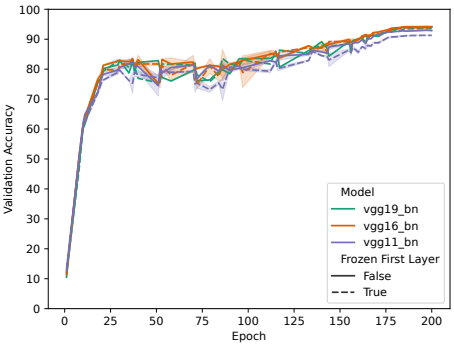
(a) Train Loss



(b) Validation Loss



(c) Train Accuracy



(d) Validation Accuracy

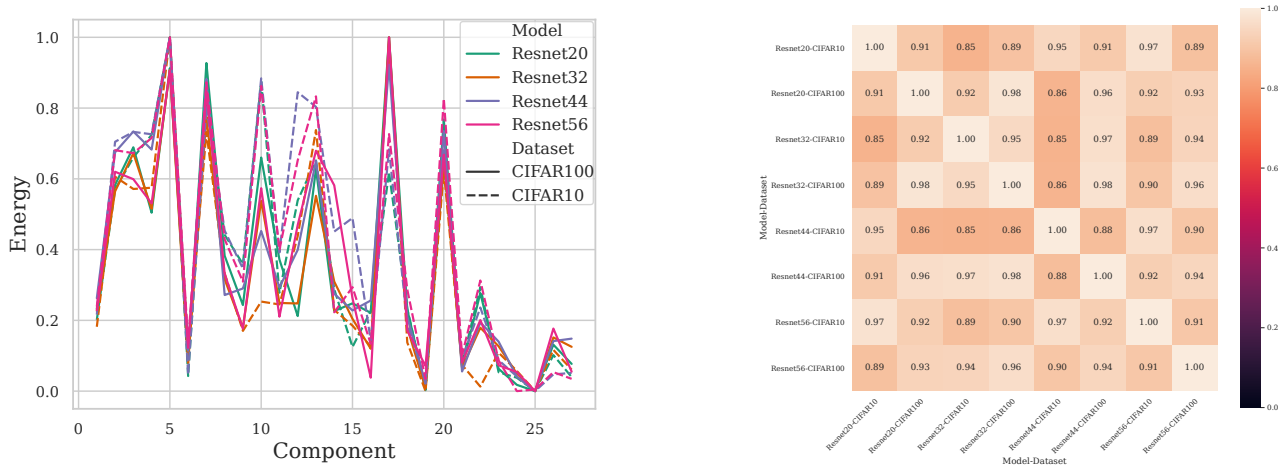
Figure 18. Loss and accuracy metrics for VGGs of different depths, with and without a frozen layer, on CIFAR10, as function of iteration.

E. Consistency for Different Datasets and Architectures

Below, are more figures portraying the high consistency between different models trained on different datasets, even when trained with random labels.

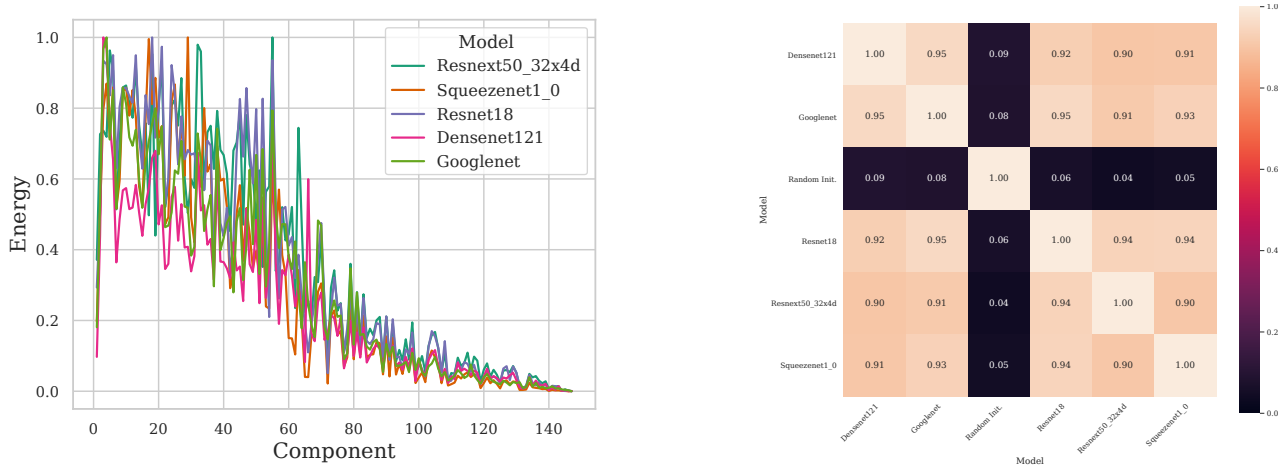
E.1. Different Datasets and Architectures

Figure 20 displays high similarity between the first layer energy profiles of models trained on ImageNet. Figure 21 and Figure 22 display high similarity for models trained on CIFAR100 and CIFAR10 respectively. Figure 19 shows different ResNets trained on either CIFAR10 and CIFAR100 learn similar profiles as well. All models in this section are pretrained models downloaded through the PyTorch model hub, from different publicly available github repositories[2].



(a) Energy Profiles (b) Correlation Coefficients

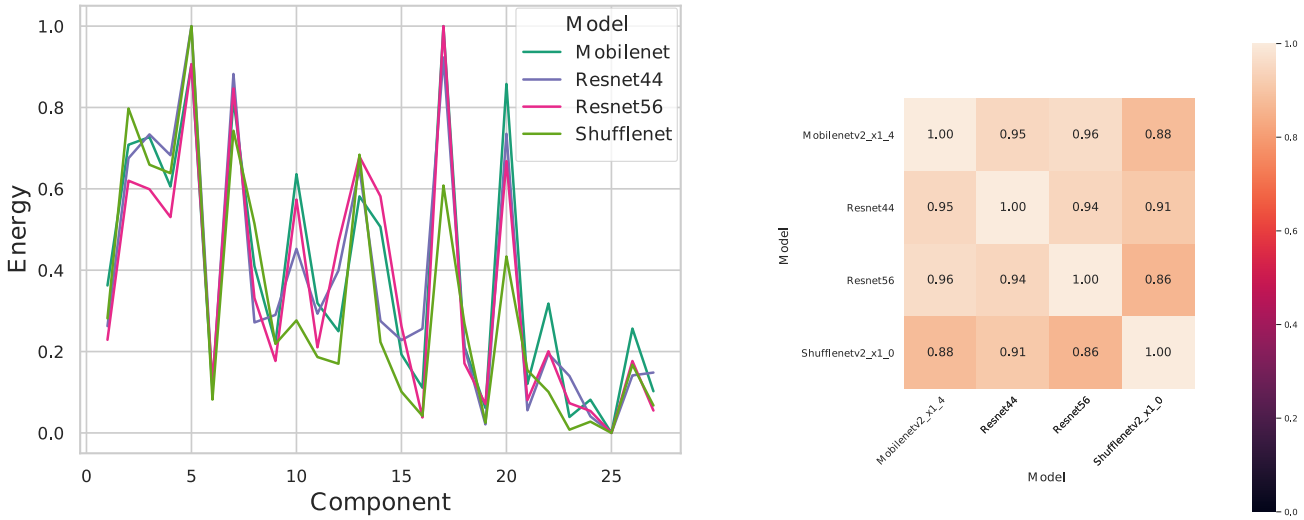
Figure 19. Different ResNets trained on different datasets all learn highly consistent energy profiles in their first layer.



(a) Energy Profiles (b) Correlation Coefficients

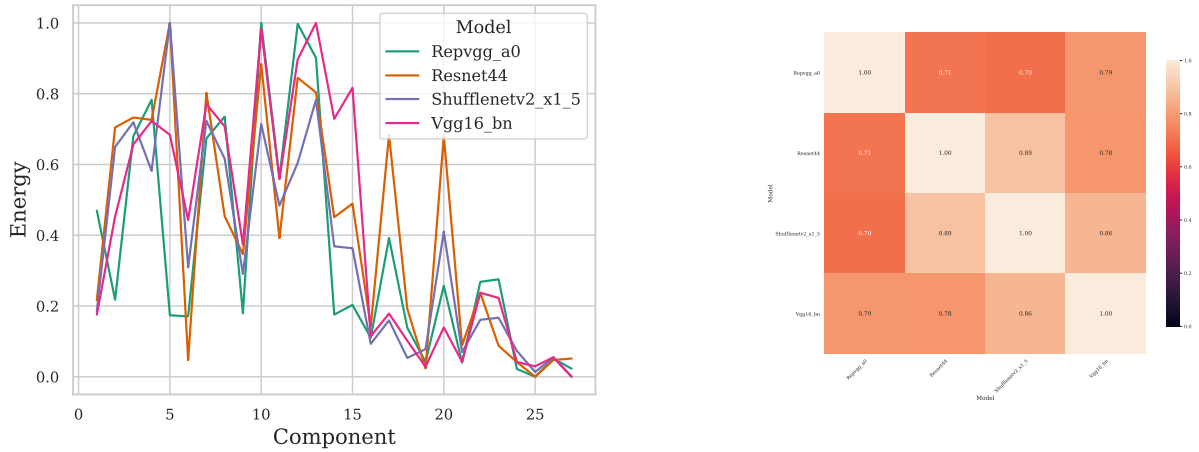
Figure 20. Energy profiles of different models trained on ImageNet with filters of dimension $3 \times 7 \times 7$. Models are generally different from a random initialization.

[2]https://github.com/chenyaofo/pytorch-cifar-models



(a) Energy Profiles (b) Correlation Coefficients

Figure 21. Energy profiles of different models trained on CIFAR100 with filters of dimension $3 \times 3 \times 3$.

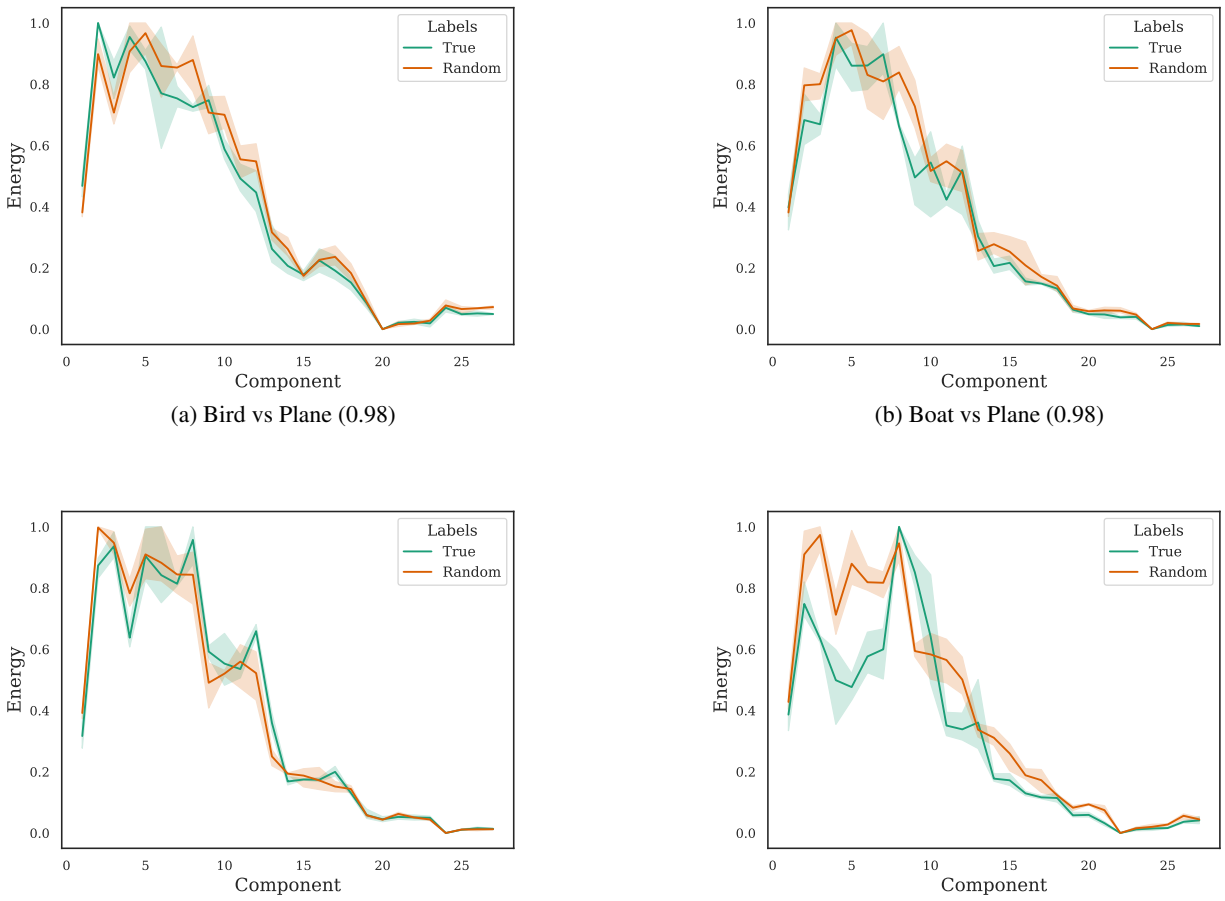


(a) Energy Profiles (b) Correlation Coefficients

Figure 22. Energy profiles of different models trained on CIFAR10 with filters of dimension $3 \times 3 \times 3$.

E.2. Consistency for True and Random Labels

Figure 23 displays high similarity between VGG11's trained on different binary subsets of CIFAR10 with true and random labels, as discussed in Section 3.



(a) Bird vs Plane (0.98)

(b) Boat vs Plane (0.98)

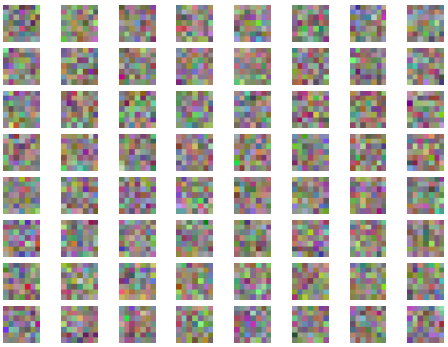
(c) Dog vs Cat (0.98)

(d) Dog vs Frog (0.92)

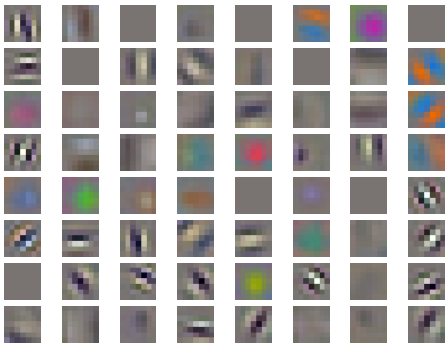
Figure 23. Energy profiles of VGG11 trained with true and random labels on different binary subsets of CIFAR10. Correlation coefficients in parentheses.

F. Visual Similarity of Filters in the First Layer

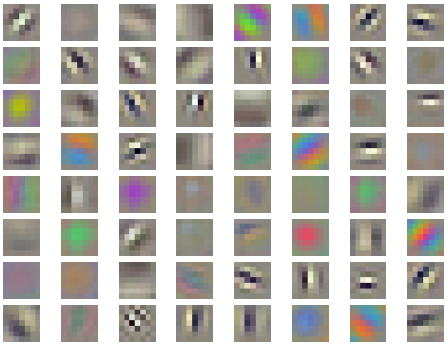
As has already been pointed out by (Krizhevsky et al., 2012; Li et al., 2015) filters learned by CNNs learn visually similar filters. For the readers convenience, Figure 24 displays filters taken from different networks trained on ImageNet. Notice these are noticably differ from a random initialization.



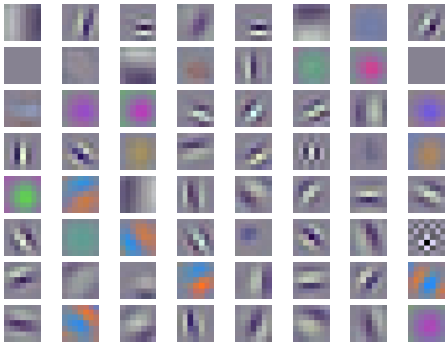
(a) Initialized ResNet18 Filters



(b) Trained ResNet18 Filters



(c) Trained GoogleNet Filters



(d) Trained DenseNet Filters

Figure 24. Different CNNs (24b, 24c, 24d) trained on ImageNet learn a highly consistent first layer despite using different architectures. These filters are very different from the initial, random filters (24a) showing that consistent representation learning has occurred.