ON REPRESENTATION LEARNING IN THE FIRST LAYER OF DEEP CNNS AND THE DYNAMICS OF GRADIENT DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

It has previously been reported that the representation that is learned in the first layer of deep CNNs is very different from the initial representation and highly consistent across initialization and architecture. In this work, we quantify this consistency by considering the set of filters as a filter bank and measuring its energy distribution. We find that the energy distribution is remarkably consistent and try to determine the source of this consistency. We show that this consistency cannot be explained by the fact that CNNs learn a representation that is useful for recognition and that CNNs trained with fixed, random filters in the first layer yield comparable recognition performance to full learning. We then show that similar behavior occurs in simple, linear CNNs and obtain an analytical characterization of the energy profile of linear CNNs trained with gradient descent. Our analysis shows that the energy profile is determined by two factors (1) the correlation of the average patch and the class label and (2) an implicit bias given the dynamics of gradient descent. Finally, we show that in commonly used image recognition datasets the correlation between the average patch and the class label is very low and it is the implicit bias that best explains the consistency of representations observed in real-world CNNs.

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; Doimo et al., 2020; Gidaris et al., 2018).

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 since each neuron can be characterized by a single linear filter which can be easily visualized as an image. Figure 1 shows examples of visualizations of the learned filters: unlike the initial filters which are random and devoid of structure, the trained filters resemble Gabor filters (Krizhevsky et al., 2012) and are visually similar for different trained networks.

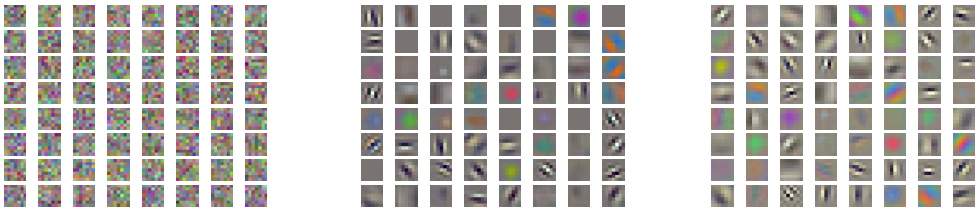
In addition to the qualitative similarity of filters that can be seen in figure 1, there have also been some reports that the filters are quantitatively similar. For example, Li et al. (2015) showed that one can often find a good match for filters learned by one CNN in the set of filters learned by another CNN. In this work we introduce a new measure for qualitatively measuring consistency of representations in the very first layer of a CNN. Using this measure, we show a remarkably high degree of consistency (correlation coefficient close to 1) between the representations that are learned by different CNNs, regardless of initializations, architectures and training sets.

The fact that these filters are so different from the initialization is interesting in the context of the theory of deep networks which indicates that under certain conditions they can be trained in a "lazy" regime (Chizat et al., 2019) - the representations in all intermediate layers hardly differing from their initialization and only the last output layer has weights that differ from initialization. Figure 1

clearly shows that "lazy training" does not occur in the first layer of deep CNNs and that consistent representation learning occurs instead.

A natural explanation for the learning of consistent filters in the first layer is that these filters are optimal in some sense for solving the recognition task. Indeed, Gabor filters and similar oriented filters were often used as a representation of images in the days of "handcrafted" features for computer vision (Dalal & Triggs, 2005). Similarly, under this explanation, the networks have simply learned that in order to minimize the training loss the first layer of deep CNNs must have filters that resemble Gabors.

In this paper we present empirical and theoretical results that are **inconsistent with this explanation**. We show that CNNs with commonly used architectures can be trained with **fixed, random filters in the first layer** and still yield comparable performance to full learning. We then show that consistent representation learning in the first layer also occurs in simple, linear CNNs and prove that for these CNNs the dynamics of gradient descent learning together with the statistics of natural image patches introduce an implicit bias towards certain filter distributions. We then show that in real-world CNNs trained on commonly used datasets, **a highly consistent representation is learned in the first layer when the true labels are replaced with random labels** and therefore that it is the implicit bias that best explains the consistency of representations observed in real-world CNNs.



(a) Initialized ResNet18 Filters (b) Trained ResNet18 Filters (c) Trained GoogleNet Filters

Figure 1: Different CNNs (1b, 1c) trained on ImageNet learn a highly consistent first layer despite using different architectures. These filters are very different from the initial, random filters showing that consistent representation learning has occured. In this paper we quantify this consistency and seek to understand its source.

2 Quantifying Consistency using Energy Profiles

The visual similarity of the filters that are learned in the first layer of CNNs (figure 1) is easy to see, but we wish to quantify the similarity of representations and go beyond the qualitative similarity. 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). In this paper we propose a new method which avoids these shortcomings and is especially relevant for the first layer of a CNN, in which the representation is a linear function of the input patch.

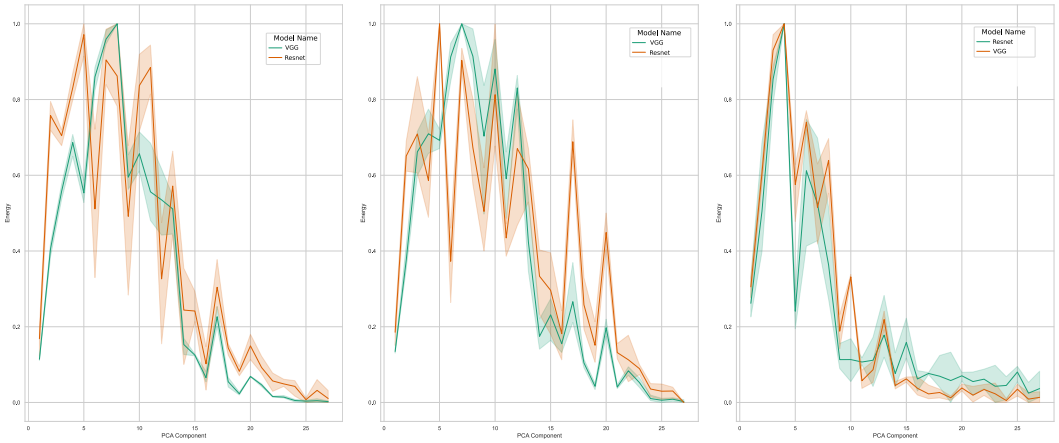
Given two patches x_1, x_2 and a linear transformation A whose rows are the filters, the squared distance between the transformed patches is $\|Ax_1 - Ax_2\|^2$ or alternatively $(x_1 - x_2)^T A^T A(x_1 - x_2)$. Thus a natural way to understand how distances are transformed when going from x_1 to Ax_1 is to look at the eigendecomposition of $A^T A$: the ith eigenvalue of $A^T A$ measures how much distances in the direction of the ith eigenvector are increased or decreased by the transformation. The eigenvectors of $A^T A$ are simply the principal components of the filters, and if we assume translation invariance of the filters, they will have the same principal components as those of natural image patches: namely sines and cosines of different spatial frequencies (Aapo Hyvärinen & Hoyer, 2009). Thus the transformation of similarities is mostly driven by the eigenvalues of $A^T A$ and we focus on these to define the consistency of learned filters.

Denote $p_1, ..., p_k$ the PCA components computed from the training images' patches and A the weights of the first layer of some model trained on these images (where each row of A, A_j^T is a filter). We define the *energy* w.r.t each component p_i:

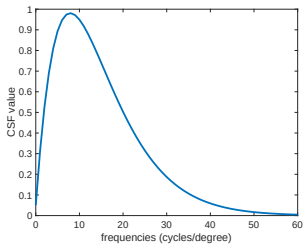
$$e_i = \|Ap_i\| = \sqrt{\sum_j (A_j^T p_i)^2} \tag{1}$$

The *energy profile* of a set of filters is simply the vector $\mathbf{e} = (e_1..e_k)$ and we measure consistency of two different sets of filters by measuring the correlation coefficient between their energy profiles. Note that this consistency measure is invariant to a rescaling of the filters, to a permutation of the filters and to any orthogonal transformation of the filters. This way of comparing linear representation is equivalent to considering the set of filters as a filter bank and measuring the sensitivity of the filter bank to different spatial frequencies.

Figures figs. 2a to 2c show 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 spatial frequencies and very low ones and the peak sensitivity is for intermediate spatial frequencies (qualitatively similar to the sensitivity pattern of the human visual system which is also sensitive to intermediate spatial frequencies, as shown in figure 2d). Table 1 quantifies this similarity. The correlation coefficient between energy profiles with different random initializations and architecture is remarkably high (over 0.98 in many cases) and the correlation between the learned profiles and the random initialization is close to zero. An expansive set of experiments on various models and datasets can be found in B.2.



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



(d) Human CSF

Figure 2: When trained with vanilla SGD on various datasets, VGG11 and ResNet18 both learn consistent energy profiles that resemble the contrast sensitivity function (CSF) of humans fig. 2d. Each line shows the average over many different initializations and the spread indicates the variance. Note the high consistency for different initializations and architectures.

Table 1: Correlation between energy profiles of VGG11, trained with different random seeds (initializations), first layer widths, over various datasets, and compared with ResNet18. It is clear that independent of architecture, seed and loss, the models trained are highly correlated while different from their random initializations.

DATASET	RANDOM SEED	TRAINED VS RANDOM INIT.	FIRST LAYER WIDTH	VGG11 VS RESNET18
CIFAR10	0.99 ± 0.004	-0.13 ± 0.18	0.98 ± 0.008	0.87 ± 0.04
CIFAR100	0.97 ± 0.01	-0.04 ± 0.04	0.98 ± 0.01	0.80 ± 0.02
CELEBA	0.99 ± 0.004	-0.18 ± 0.13	0.98 ± 0.006	0.92 ± 0.02

Thus the use of our new measure allows us to quantitatively show that deep CNNs trained with gradient descent using standard parameters do not exhibit "lazy" training in the first layer and that highly consistent representation learning takes place. 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) (see figure 2d), 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, Gabors of intermediate frequencies are 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 3 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 validation loss. 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 that consistent Gabors of intermediate frequencies would be found. Equally good training loss can be obtained with random filters in the first layer.

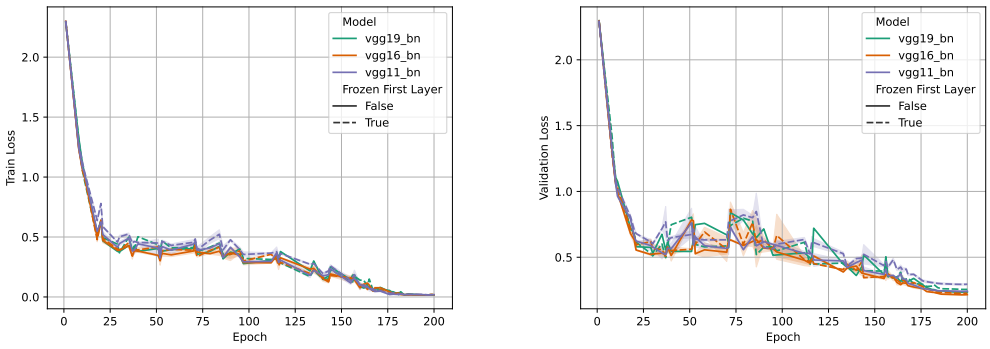
To summarize, while quantitatively highly consistent representations are learned in the first layer of commonly used CNNs, this cannot be explained by the networks minimization of the training loss, This motivates us to analyze representation learning in much simpler CNNs.

4 SIMPLE, LINEAR CNN

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. Specifically, in this simple model, the first layer includes convolutions with W 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 we use it because it allows closed form-analysis and also exhibits some of the same consistency behaviors that we found in real-world CNNs. Specifically we have found that:

- The energy profiles of simple, linear CNNs are highly consistent across initializations and widths and are very different from the energy profiles of the initial conditions.



(a) Train Loss

(b) Validation Loss

Figure 3: Training and validation loss of VGGs of different depths on CIFAR10 (with crossentropy loss) as function of iteration with frozen first layer and without. For deep networks the performance is the same with frozen layer (and see appendix for accuracy figures).

- The energy profile of simple, linear CNNs trained with gradient descent is different from the energy profile of the filters that globally optimize the loss.
- The energy profile of simple, linear CNNs are highly consistent when the true labels are replaced with random labels.

These properties are all displayed in figure 4 where we show results of training a linear model on binary tasks from CIFAR10. In all cases, the energy profile that is learned with true labels (red) is different from the initial conditions and is sensitive mostly to intermediate frequencies while the optimal energy profile (shown in blue) is quite different and shows a high sensitivity to high spatial frequencies. Training these networks with random labels give an energy profile (in green) that is similar to that of the true labels.

The following theorems show the same behaviors analytically.

Theorem 4.1. *Consider a linear CNN with arbitrary width that solves a binary classification problem with L2 loss. The energy profile for the filters that globally minimize the loss is given by* μ^2_{opt} *with:*

$$\mu_{opt} = (K^T K)^{-1} K^T y$$

Where K *is the matrix of average patches in each image (in the PCA basis) and* y *is the vector of labels.*

Proof. This simply follows from the model being equivalent to linear model in the average image patch (lemmas A.1 and A.2), and solving the suitable linear regression problem. □

Theorem 4.2. *Consider a linear CNN with arbitrary width that solves a binary classification problem with L2 loss and is trained with gradient descent starting with zero mean filters and covariance* $\sigma^2 I$*. The squared energy profile for the filters at any iteration is given by* $\mu^2_{GD} + \sigma^2$ *with:*

$$\mu_{GD} = (K^T K + \Lambda)^{-1} K^T y$$

With the same K *and* y *as in theorem 4.1, and* Λ *is a spectral regularizer that depends on the learning rate, number of iterations and* $K^T K$*.*

Proof. The full proof is given in the appendix and uses a similar technique that was used to derive spectral biases in gradient descent learning of fully connected networks LeCun et al. (1991). See A.3 for a full proof. □

Theorem 4.3. *Consider a linear CNN with arbitrary width that solves a binary classification problem with L2 loss and is trained with gradient descent starting with zero mean filters and covariance*

$\sigma^2 I$. *If the label y is uncorrelated with the average patch in each image then the squared energy profile for the filters at any iteration is given by $\mu_0^2 + \sigma^2$ with:*

$$\mu_0 \propto (K^T K + \Lambda)^{-1} K^T 1 \tag{2}$$

Where K is the matrix of average patches in each image and 1 is a vector of all ones and Λ is a spectral regularizer that depends on the learning rate and number of iterations.

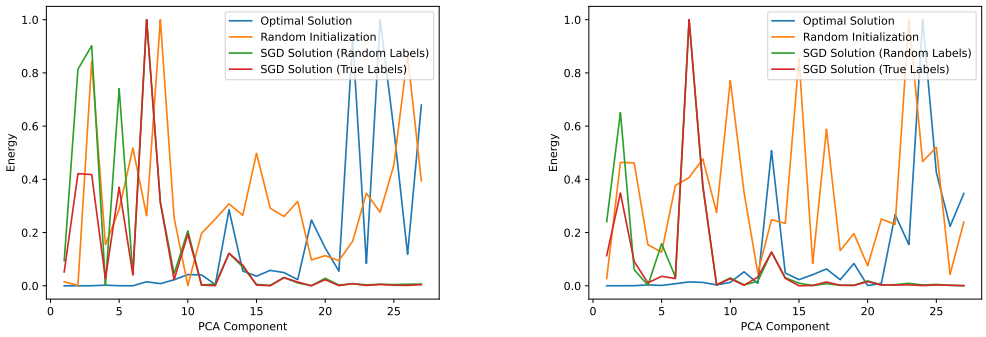
Proof. This follows from theorem 4.2 and the fact that the quantity Ky is proportional to the empirical expectation of the product between each average image patch and it's label. When the average patch is uncorrelated with the label, this expectation is the product of the expected average patch and expected label, so it is proportional to $K^T 1$ which is the expectation of the average patch. □

Corollary 4.4. *Consider a linear CNN with arbitrary width that solves a binary classification problem with L2 loss and is trained with gradient descent starting with zero mean filters and covariance $\sigma^2 I$. If the average patch is the same in the two classes, then training with the true labels and training with random labels will give the same energy profile given by:*

$$\mu_0 \propto (K^T K + \Lambda)^{-1} K^T 1$$

Proof. Follows from $K^T y$ being a sum of all average image patches with $y_i = 1$. If both average class patches are equal, then in expectation over a random (binary) y, the sums of all average patches will be equal. Full description of μ_0 can be found in theorem A.5. □

These theorems show that in the case of simple, linear CNNs different networks (initial conditions, widths) will learn the same representation in the first layer but despite the fact that the loss is convex, the learned representation will **not** be the representation that globally optimizes the training loss with any finite number of training steps. Rather, the use of gradient descent will introduce an implicit bias that favors certain energy profiles that depend on the number of training steps and the learning rate (with the form of the bias given explicitly by equation 2). This implicit bias causes the learned profiles to be highly consistent yet very different from the optimal one.



(a) Boat vs Plane

(b) Dog vs Frog

Figure 4: Training a Linear CNN on the same 2-class subsets of CIFAR10 as in fig. 6 results in an almost identical energy profile for true and random labels (correlation over 0.9), as almost all peaks in the energy profiles align. On the other hand, this representation is different from the optimal one (correlation of under -0.2) for recognition obtained by solving a linear regression problem. For more correlation coefficient see 3.

5 Implicit Bias in Nonlinear CNNs

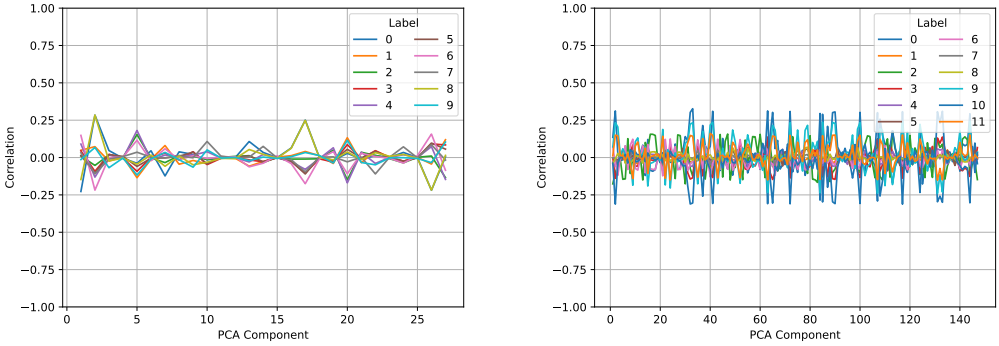
The theoretical analysis of linear CNNs shows that if the true labels are uncorrelated with the average patch in an image, the learned energy profile will converge to a consistent profile that is determined

Table 2: Correlation between energy profiles of VGG11 trained with true and random labels for different datasets as depicted in fig. 6. We show the average correlation (over multiple random initializations and firs layer widths) and the standard deviations. It is clear that the first layer is highly correlated when training with true and random labels.

DATASET	VGG (TRUE) VS INIT.	VGG (RANDOM) VS INIT.	VGG (RANDOM) VS VGG (TRUE)
CIFAR10	-0.13 $\pm$ 0.17	0.03 $\pm$ 0.22	**0.90** $\pm$ **0.02**
CIFAR100	-0.044 $\pm$ 0.04	0.14 $\pm$ 0.13	**0.91** $\pm$ **0.01**
CELEBA	-0.18 $\pm$ 0.13	0.085 $\pm$ 0.07	**0.96** $\pm$ **0.03**

by the dynamics of gradient descent and the statistics of the input patches. We therefore ask: is the consistent energy profile that we find in real-world CNNs also due to the dynamics of SGD?

According to our analysis, the implicit bias is strongest when the label is uncorrelated with the average patch. We measured this correlation in commonly used image classification datasets by computing the correlation coefficient between the average PCA value in an image and its label for different tasks (a binary classification of one class versus the rest). The results are shown in figure 5. For almost all tasks and coefficients, the correlation coefficient is close to zero.



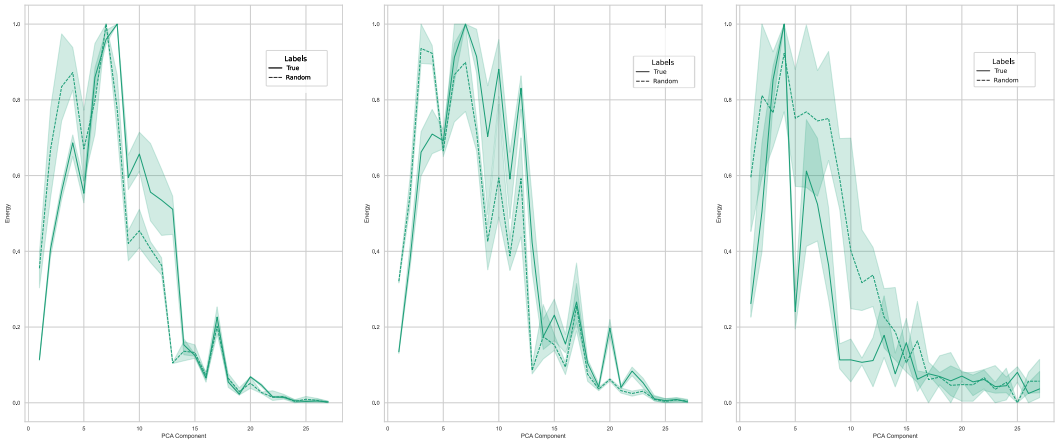
(a) CIFAR10 (b) ImageNet (12)

Figure 5: Correlation between the projection of average image patches onto the PCA basis and the class labels for CIFAR10 (fig. 5a) and a 12 class subset of ImageNet (fig. 5b). Correlations are all around 0, suggesting the average patch assumption is true for real datasets.

Given the small amount of correlation, we would expect a similar energy profile when we train with random labels and true labels. Maennel et al. (2020) have already shown that when CNNs are trained with random labels, the representations that are learned in the first layers are still useful for other tasks. Here, we ask a more quantitative question: are the energy profiles the same?

As shown in figures 6 and table 2 the answer is clearly yes. Correlations above 0.9 are consistently observed even when the true labels are replaced with random labels and the representations that are learned are still mostly sensitive to intermediate spatial frequencies. This is true both when trained on multiclass recognition problems (e.g. CIFAR10, CIFAR100, CelebA) and when trained on smaller, 2-class problems for which we've already seen consistency of linear CNNs (fig. 4).

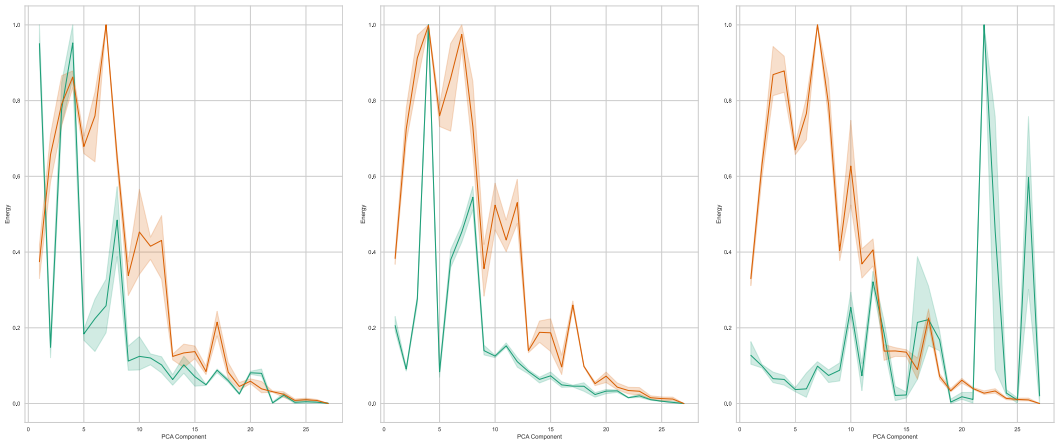
As an additional test of the hypothesis that the energy profiles we see in real-world CNNs are mostly due to the implicit bias, we created new datasets in which we artificially created strong correlations between the label and particular PCA components and trained VGG on them. The image labels were determined by the average patch projection onto some PCA component, such that the 5,000 images with the *largest* magnitude of projection were labeled with 1 and so on. As can be seen in fig. 7, once changing the average patch of each class manually the correlation between the true



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

Figure 6: VGG11 trained on CIFAR10 (fig. 6a) CIFAR100 (fig. 6b) and a CelebA classification task (fig. 6c) exhibit similar energy patterns when trained with true and random labels. These are also highly correlated and differ from initialization (see table 2). Further experiments on binary CIFAR10 subsets and with ResNet can be found in B.4

and random profiles decreases from the original 0.9 ± 0.02 to as low as -0.24 ± 0.02, depending on the component changed and the learned energy profile no longer resemble the human sensitivity function.



(a) PCA Component 2 (0.63 ± 0.02) (b) PCA Component 4 (0.77 ± 0.01) (c) PCA Component 24 (-0.24 ± 0.03)

Figure 7: Energy profile of the first layer when labeling images of CIFAR10 based on their total projection on different patch-PCA components, trained with true (green line) and random (orange line) labels. Creating a correlation between the label and the PCA component causes the first layer to learn this component, depicted by a spike in the energy. Mean correlation values are in parenthesis.

6 Related Work

The fact that different CNNs tend to learn similar filters in the first layer has been reported previously (e.g. Yosinski et al. (2014); Sarwar et al. (2017); Luan et al. (2017); Alekseev & Bobe (2019)), 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.

Prior works have also studied the ability of neural networks to overfit random labels (Arpit et al., 2017), and use representations learned in this regime for transfer learning. Maennel et al. (2020) hypothesised that the ability of networks trained on random labels to transfer to new tasks is due to the fact that under certain conditions, the first layers of networks trained with random labels have aligned covariances with input image patches. We expand on this hypothesis by showing that networks trained with true labels, for which there is no alignment guarantee, display the same energy profile as networks trained with random labels. We show this quantitatively for VGG11 and ResNet and theoretically for linear CNNs with a single hidden layer.

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 conditions, 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 non-linear networks are biased towards learning lower frequencies first (Rahaman et al., 2019). Our work follows this line, focusing on the features learned in the first layer as a result of this bias, and that of the input image statistics.

In its theoretical part, our analysis methods follow closely on the methods used by (LeCun et al., 1991; Hacohen & Weinshall, 2022) which analyze the dynamics of weights in a fully connected network during learning with gradient descent. We use a similar technique but our focus is on the first layer of a CNN. Additionally, we rely on linear networks to gain insight into the behavior of nonlinear networks, following previous works (Hacohen & Weinshall, 2022; **?**; Gissin et al., 2019). In the same manner, we support our simplified theoretical claims by quantitatively showing consistency of the theory in real-world CNNs such as VGG.

As a result of the consistency of Gabors being learned in the first layers of CNNs such as ResNet, GoogLeNet and DenseNet, each a state-of-the-art at its time - some lines of work attempted building CNNs with learnable Gabors Sarwar et al. (2017); Luan et al. (2017); Alekseev & Bobe (2019) in the first layer. Nevertheless, these failed to reach the high level of performance on benchmark tasks such as the "vanilla" architectures. Our work expands on this contradiction by showing that not only do the networks consistently learn Gabor filters but also a specific distribution of their frequencies.

The distribution mentioned above, was portrayed in our work using the *energy profile* of the first layers. This measure follows a line of many works in the field of measuring and visualizing similarities between representations Csiszárik et al. (2021); Kornblith et al. (2019); Nguyen et al. (2021); Li et al. (2015); Doimo et al. (2020), varying between comparing the output of transformations induced by the neurons or the neurons themselves. The energy profile is yet another method in this line, while allowing for semantically meaningful (as PCA components correspond to spatial frequencies) visualization of the representation, without any need for dimensionality reduction.

7 Discussion

The dramatic success of CNNs in computer vision has lead to increased interest in the representations that they learn. 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 and architectures. We have examined the hypothesis that this consistency is due to networks learning a representation that is useful for object recognition and presented results that are inconsistent with that hypothesis. By analysing a simple, linear CNNs we have shown that such networks will provably converge to a consistent energy profile under many conditions, but this profile may have nothing to do with the labels and is instead determined by an implicit bias due to the dynamics of gradient descent and the statistics of the input patches.

REFERENCES

Jarmo Hurri Aapo Hyvärinen and Patrick O. Hoyer (eds.). *Natural Image Statistics*. Springer-Verlag London, London, UK, 2009.

Andrey Alekseev and Anatoly Bobe. 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.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh (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.

Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Jacob Abernethy and Shivani Agarwal (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.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations, 2021.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 886–893 vol. 1, 2005. doi: 10.1109/CVPR.2005.177.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity with statistical testing. *arXiv preprint arXiv:2108.01661*, 2021.

Diego Doimo, Aldo Glielmo, Alessio Ansuini, and Alessandro Laio. Hierarchical nucleation in deep neural networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL http://arxiv.org/abs/1803.07728.

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

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

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

Guy Hacohen and Daphna Weinshall. 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.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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.

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

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

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar (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.

Shangzhen Luan, Baochang Zhang, Chen Chen, Xianbin Cao, Jungong Han, and Jianzhuang Liu. Gabor convolutional networks. *CoRR*, abs/1705.01450, 2017. URL http://arxiv.org/abs/1705.01450.

Hartmut Maennel, Ibrahim Alabdulmohsin, Ilya Tolstikhin, Robert J. N. Baldock, Olivier Bousquet, Sylvain Gelly, and Daniel Keysers. What do neural networks learn when trained with random labels? 2020. URL https://arxiv.org/abs/2006.10455.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. 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.

Cynthia Owsley. 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.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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.

Syed Shakib Sarwar, Priyadarshini Panda, and Kaushik Roy. Gabor filter assisted energy efficient fast learning convolutional neural networks. *CoRR*, abs/1705.04748, 2017. URL http://arxiv.org/abs/1705.04748.

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

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

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

APPENDIX A LINEAR CONVOLUTIONAL NETWORKS

A.1 PROOFS OF THEOREMS ON LINEAR CONVOLUTIONAL NETWORKS

We first begin with a basic claim on the model composed of a hidden convolutional layer and followed by a global average pool.

Lemma A.1. *A linear CNN of depth 1 (followed by a global average pool) trained with MSE loss, is equivalent to linear regression on the average image patch.*

Proof. Let $\{X_i\}_{i=1}^N$ with $X_i \in \mathbb{R}^{c\times w\times h}$ be the set of training images, $\{y_i\}_{i=1}^N$ their binary labels ($y_i \in \{0,1\}$) and the weights of the first layer be $W \in \mathbb{R}^{k\times c\times d\times d}$ - k filters of dimension $d\times d$. Denote the output dimensions of the convolution as w', h', then:

$$\mathcal{L}\left(W;\{(X_i,y_i)\}_{i=1}^N\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}||(\frac{1}{k\cdot w'\cdot h'}\sum_{k,w',h'} X_i * W) - y_i||^2$$

Summing over the output dimensions is equivalent to summing over a dot product of the patches with a single filter, therefore denoting $K_i \in \mathbb{R}^{w'\cdot h'\times c\cdot d^2}$ as the patch matrix of the i'th image and $\tilde{W} \in \mathbb{R}^{c\cdot d^2\times k}$ the reshaped weights matrix:

$$\mathcal{L}\left(W;\{(X_i,y_i)\}_{i=1}^N\right) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|\left(\frac{1}{w'\cdot h'}\mathbf{1}\right)^T K_i\tilde{W}\left(\frac{1}{k}\mathbf{1}\right) - y_i\right\|^2$$

And noting that $\left(\frac{1}{w'\cdot h'}\mathbf{1}\right)^T K_i$ is the average patch of the i'th image and $\tilde{W}\left(\frac{1}{k}\mathbf{1}\right)$ the average filter concludes the proof. □

Another lemma we'll use further on claims that during training of the linear CNN model, only the average filter changes while the filter covariance remains as during initialization. Therefore proofs from one filter to multiple filters are easily extendable.

Lemma A.2. *In a linear CNN of depth 1 followed by a global average pool, of any width, trained with GD and MSE loss, the average filter changes during iterations while the covariance of filters remains as during initialization.*

Proof. Following the notation of lemma A.1, denote $K \in \mathbb{R}^{N\times c\cdot d^2}$ as the average image patch matrix - the image whose i'th row is the average patch of the image X_i, and the network consists of filters $w_1....w_m$, therefore trained with the following loss:

$$\mathcal{L}\left(w_1...w_m;K,y\right) = \left\|\frac{1}{m}\sum_{i=1}^{m}Kw_i - y\right\|^2 \tag{3}$$

$$= \left\|K\left(\frac{1}{m}\sum_{i=1}^{m}w_i\right) - y\right\|^2 = \left\|K\bar{w} - y\right\|^2 \tag{4}$$

Where $\bar{w}$ is the average filter. The dynamics of a single filter in this layer:

$$\frac{\partial\mathcal{L}}{\partial w_j} = \frac{1}{2m}K^T\left(K\left(\frac{1}{m}\sum_{i=1}^{m}w_i\right) - y\right) = \frac{1}{2m}K^T\left(K\bar{w} - y\right) \tag{5}$$

Meaning that the gradients w.r.t to all filters are equal and depend only on the average filter at the current iteration.

By recursion we can see that the change in the average filter is as follows, for learning rate η:

$$\bar{w}^t = \frac{1}{m}\sum_{i=1}^{m}\left(w_i^{t-1} - \eta\nabla\mathcal{L}^{t-1}(w_i)\right) = \frac{1}{m}\sum_{i=1}^{m}\left(w_i^{t-1} - \eta\nabla\mathcal{L}^{t-1}(\bar{w})\right) \tag{6}$$

$$= \left(\frac{1}{m}\sum_{i=1}^{m}w_i^{t-1}\right) - \eta\nabla\mathcal{L}^{t-1}(\bar{w}) = \bar{w}^{t-1} - \eta\nabla\mathcal{L}^{t-1}(\bar{w}) \tag{7}$$

Concluding that the gradients for all filters are equal, and depend only on the average filter. □

Theorem A.3. *Let K be a matrix whose i'th row is the average image patch of the i'th image and y is a vector with the labels of all images, and let $\bar{K} = KU$ be the same matrix in the PC basis (with U being the PCA eigenvector matrix).* ***The squared energy profile of weights of linear CNN****, initialized with random weights sampled zero mean and covariance $\sigma^2 I$ and trained with GD, is equal to the following:*

$$e_i^2 := \frac{1}{M}\sum_{j=1}^{M}\langle f_j, p_i\rangle^2 = \tilde{w}_i^2 + \sigma^2 \tag{8}$$

where $\tilde{w} = \left(\bar{K}^T\bar{K} + \Lambda\right)^{-1}\bar{K}^T y$ is the solution to a regularized regression problem in the PC basis, that regresses the average patch *in an image with its label, with $\Lambda = \Lambda(K^TK, t, \eta)$ a matrix depending on the eigenvalues of K^TK, the iteration of GD and the step size.*

Proof. It follows from lemma A.2 that during training all filters change by the average filter. We'll show that a single filter (at iteration t of GD) corresponds to a solution to ridge regression with some matrix $\Lambda = \Lambda(t, \eta, K^TK)$ with η being the step size. Opening the recursion of GD updates, and assuming w is initialized at $w = 0$:

$$\begin{aligned} w_t &= w_{t-1} - \eta\bar{K}^T\left(\bar{K}w_{t-1} - y\right) = w_{t-1} - \eta\bar{K}^T\bar{K}w_{t-1} + \eta\bar{K}^T y \\ w_t &= \eta\sum_{j=0}^{t-1}\left(I - \eta\bar{K}^T\bar{K}\right)^j\bar{K}^T y \end{aligned} \tag{9}$$

In this coordinate system, $\bar{K}^T\bar{K}$ is a diagonal matrix with the empirical variances $\hat{\sigma}_i^2$ on the diagonal if it is centered. If the matrix isn't centered, then $\bar{K}^T\bar{K} = \hat{\Sigma} + \hat{\mu}\hat{\mu}^T$ where $\hat{\Sigma}$ is a diagonal matrix with the empirical variances on the diagonal and $\hat{\mu}_i$ is the empirical mean estimating $\mathbb{E}_x\left[\langle x, p_i\rangle\right]$. This is because in PCA coordinates, $\bar{K} = KU$, where U contains the eigenvectors as columns. Since K isn't centered, $K = K_0 + 1K_{avg}^T$ with K_0 being zero meaned and K_{avg} being the average row. Therefore, $\bar{K}^T\bar{K} = \left(K_0U + 1K_{avg}^TU\right)^T\left(K_0U + 1K_{avg}^TU\right) = \hat{\Sigma} + \hat{\mu}\hat{\mu}^T$, where the phrase $K_0^T(1K_{avg}^T)$ disappears since K_0 has zero mean. Therefore:

$$w_t = \eta\sum_{j=0}^{t-1}\left(I - \eta\bar{K}^T\bar{K}\right)^j\bar{K}^T y = \eta\sum_{j=0}^{t-1}\left(I - \eta\left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^T\right)\right)^j\bar{K}^T y \tag{10}$$

Notice that $\left(I - \eta\left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^T\right)\right)^j$ can be decomposed in the following manner using the binomial theorem:

$$\left(I - \eta\left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^T\right)\right)^j = \left(I - \eta\hat{\Sigma}\right)^j + \sum_{k=1}^{j}\binom{j}{k}(-\eta)^k\left\|\hat{\mu}\right\|^{2(k-1)}\left(I - \eta\hat{\Sigma}\right)^{j-k}\hat{\mu}\hat{\mu}^T \tag{11}$$

Putting it back into equation 10:

$$\begin{aligned} w_t &= \eta\sum_{j=0}^{t-1}\left(I - \eta\left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^T\right)\right)^j\bar{K}^T y \\ &= \eta\left(\sum_{j=0}^{t-1}\left(I - \eta\hat{\Sigma}\right)^j + \sum_{k=1}^{j}\binom{j}{k}(-\eta)^k\left\|\hat{\mu}\right\|^{2(k-1)}\left(I - \eta\hat{\Sigma}\right)^{j-k}\hat{\mu}\hat{\mu}^T\right)\bar{K}^T y \end{aligned} \tag{12}$$

Looking at the i'th coordinate, with λ_i being the i'th eigenvalue on the diagonal of $\hat{\Sigma}$:

$$\begin{aligned} w_t(i) =& (\bar{K}^T y)(i)\sum_{j=0}^{t-1}(1 - \eta\lambda_i)^j \\ &+ (\hat{\mu}\hat{\mu}^T\bar{K}^T y)(i)\sum_{j=1}^{t-1}\frac{1}{\left\|\hat{\mu}\right\|^2}\left(\sum_{k=1}^{j}\binom{j}{k}(-\eta\left\|\hat{\mu}\right\|^2)^k\left(I - \eta\lambda_i\right)^{j-k}\right) \end{aligned} \tag{13}$$

After some algebra:

$$w_t(i) = \left(\frac{1 - \left(1 - \eta(\lambda_i + \|\hat{\mu}\|^2)\right)^t}{\|\hat{\mu}\|^2 (\lambda_i + \|\hat{\mu}\|^2)} - \frac{1 - (1 - \eta\lambda_i)^t}{\lambda_i} \right) (\hat{\mu}\hat{\mu}^T \bar{K}^T y)(i) + \left(\frac{1 - (1 - \eta\lambda_i)^t}{\lambda_i} \right) (\bar{K}^T y)(i) \tag{14}$$

And in matrix notation, define the diagonal matrix A with $A_{ii} = \frac{1-(1-\eta\lambda_i)^t}{\lambda_i}$ as the i'th element on the diagonal, and the diagonal matrix B with $B_{ii} = \frac{1-\left(1-\eta(\lambda_i+\|\hat{\mu}\|^2)\right)^t}{\|\hat{\mu}\|^2(\lambda_i+\|\hat{\mu}\|^2)}$ the i'th element on the diagonal and we get that:

$$w_t = (B - A)\hat{\mu}\hat{\mu}^T \bar{K}^T y + A\bar{K}^T y \tag{15}$$

Solving the following:

$$w_t = \left(\bar{K}^T \bar{K} + \Lambda\right)^{-1} \bar{K}^T y = \left(\hat{\Sigma} + \hat{\mu}\hat{\mu}^T + \Lambda\right)^{-1} \bar{K}^T y \tag{16}$$

we get that:

$$\Lambda = \left(B + (A - B)\hat{\mu}\hat{\mu}^T\right)^{-1} - \hat{\Sigma} - \hat{\mu}\hat{\mu}^T \tag{17}$$

and we got a definition for the regularization matrix.

Since the filter covariance stays constant throughout training due to lemma A.2, treating the filters as a random variable initialized with covariance $\sigma^2 I$ (in PCA basis) means that their empirical second moment is equal to the sum of the squared mean and variance. Therefore denoting the filters in PCA basis as $\tilde{f}_j$, we get that in the ith coordinate:

$$\frac{1}{M}\sum_{j=1}^{M} \langle f_j, p_i \rangle^2 = \left(\frac{1}{M}\sum_{j=1}^{M} \langle f_j, p_i \rangle \right)^2 + \frac{1}{M}\sum_{j=1}^{M} \left(\langle f_j, p_i \rangle - \left(\frac{1}{M}\sum_{j=1}^{M} \langle f_j, p_i \rangle \right) \right)^2 = \tilde{w}^2(i) + \sigma^2 \tag{18}$$

□

Theorem A.4 (Effect of Labels)**.** *Let W^t_{True} be the weights of the first layer of a linear CNN with a single hidden layer and any width, trained for t steps on a binary classification task with MSE loss and gradient descent, and let W_{Random} be the weights of the first layer of the same CNN trained with **random** labels drawn from a Bernoulli distribution. If the average patch of both classes is identical, and the dataset is balanced between them, then at any training iteration:*

$$\mathbb{E}_{y \sim \textit{Bernoulli}\left(\frac{1}{2}\right)} \left[W^t_{\textit{Random}} \right] = W^t_{\textit{True}} \tag{19}$$

Proof. Let $K \in \mathbb{R}^{N \times c \cdot d^2}$ be the average image patch matrix and $y \in \{0, 1\}^N$ the image labels. From lemma A.1, training a linear CNN with 1 layer followed by a global average pool is equivalent to solving the following linear regression problem for weights matrix $W \in \mathbb{R}^{c \cdot d^2 \times 1}$:

$$\mathcal{L}(W; K, y) = \frac{1}{N}\frac{1}{2} \|KW - y\|^2$$

Using gradient descent with learning rate η, the update rule for W is:

$$W_t = W_{t-1} - \frac{\eta}{N}\left(K^T(KW_{t-1}y)\right) = \left(I - \frac{\eta}{N}K^T K\right) W_{t-1} + \frac{\eta}{N}K^T y \tag{20}$$

Notice that in expectation, $\mathbb{E}_{y \sim \text{Bernoulli}\left(\frac{1}{2}\right)}[y] = \frac{1}{2}\mathbf{1}$, therefore $\mathbb{E}_{y \sim \text{Bernoulli}\left(\frac{1}{2}\right)}\left[K^T y\right]$ is (half) the sum of all average image patches. From our assumption, the average image is equal between classes.

Denote this average patch as z, and since K is the average patch matrix $z = \frac{2}{N}Ky$. Combining this observation with the above:

$$\mathbb{E}_{y\sim\text{Bernoulli}\left(\frac{1}{2}\right)}\left[\frac{\eta}{N}K^T y\right] = \frac{\eta}{N}\cdot\frac{1}{2}K^T\mathbf{1} = \eta\frac{1}{2N}N\cdot z = \frac{\eta}{N}K^T y \tag{21}$$

And that concludes the proof. Note that we assumed that the CNN is of width 1, but using lemma A.2 is enough for generalizing to any width. □

Theorem A.5 (Solution in PCA Basis)**.** *Let $\tilde{w} = \left(\bar{K}^T\bar{K} + \Lambda\right)^{-1}\tilde{K}^T y$ be as described in theorem A.3, for $\bar{K}$ the average image patch matrix in the PCA basis and $\Lambda = \Lambda(\bar{K}^T\bar{K}, t, \eta)$. Denote $\hat{\mu}$ as the empirical mean projection onto the PCA basis and $\hat{\Sigma}$ as the the uncentered data covariance in PCA basis. If the labels are drawn randomly from a Bernoulli distribution, then in expectation, $\tilde{w}$ can be calculated at any iteration t and for any step size η with the following formula:*

$$\mathbb{E}_{y\sim\textit{Bernoulli}\left(\frac{1}{2}\right)}\left[\tilde{w}\right] \propto \left(I - \frac{\hat{\Sigma}'^{-1}\mu\mu^T}{1+\mu^T\hat{\Sigma}'^{-1}\mu}\right)\hat{\Sigma}'^{-1}\mu \tag{22}$$

with $\hat{\Sigma}' = \hat{\Sigma} + \Lambda$.

Proof. Following the notation from before, denote $K \in \mathbb{R}^{N\times c\cdot d^2}$ as the average patch matrix, and $\bar{K}$ as the same matrix in the PCA basis coordinates. From theorem A.3 $\bar{K}^T\bar{K} = \hat{\Sigma} + \hat{\mu}\hat{\mu}^T$. Solving the linear ridge regression problem in this coordinate system as described in theorem A.3:

$$L(w;\bar{K},y) = \frac{1}{2}\left\|\bar{K}w - y\right\|^2 + \frac{1}{2}w^T\Lambda w \Rightarrow \bar{w} = \left(\bar{K}^T\bar{K} + \Lambda\right)^{-1}\bar{K}^T y \tag{23}$$

In expectation over a random y, as described in theorem A.4: $\mathbb{E}\left[y\right] = \frac{1}{2}\mathbf{1}$, therefore $\mathbb{E}\left[\tilde{K}^T y\right] = \frac{N}{2}\hat{\mu}$.

As mentioned before, $\bar{K}^T\bar{K} = \hat{\Sigma} + \hat{\mu}\hat{\mu}^T$. Define $\hat{\Sigma}' = \hat{\Sigma} + \Lambda$ a matrix summing the PCA variances and the regularization coefficients. Now using Woodbury matrix identity:

$$\left(\hat{\Sigma}' + \mu\mu^T\right)^{-1} = \hat{\Sigma}'^{-1} - \hat{\Sigma}'^{-1}\mu(I + \mu^T\hat{\Sigma}'^{-1}\mu)\mu^T\hat{\Sigma}'^{-1} = (I - \frac{\hat{\Sigma}'^{-1}\mu\mu^T}{1+\mu^T\hat{\Sigma}'^{-1}\mu})\hat{\Sigma}'^{-1}$$

and we get that:

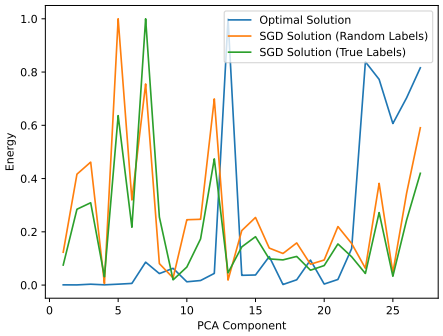
$$\tilde{w} \propto (I - \frac{\hat{\Sigma}'^{-1}\mu\mu^T}{1+\mu^T\hat{\Sigma}'^{-1}\mu})\hat{\Sigma}'^{-1}\mu$$

□

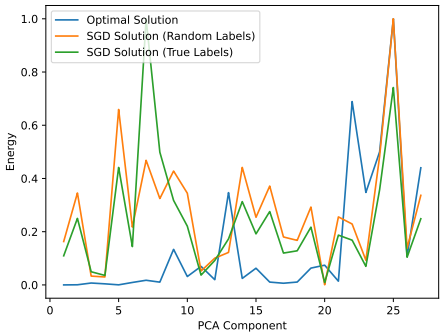
A.2 Correlation Figures

As mentioned in 4, energy profiles of linear CNNs had much higher similarity when training with true and random labels using SGD, compared to their random initialization and the optimal solution to the corresponding linear regression problem.

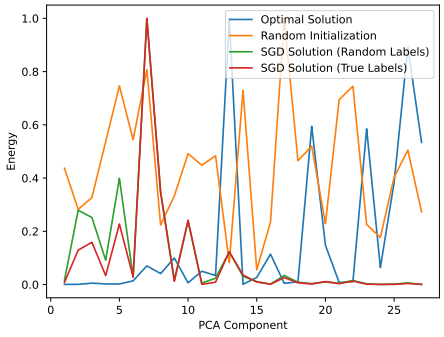
To complement 4, 3 displays the mean and standard deviation of correlation coefficients between the mentioned energy profiles. Again, it is clear that there is a high similarity in energy profiles when training a linear CNN with SGD on true and random labels.



(a) Car vs Truck



(b) Dog vs Cat



(c) Bird vs Plane

Figure 8: More examples of training a Linear CNN on 2-class subsets of CIFAR10.For correlation coefficient see 3.

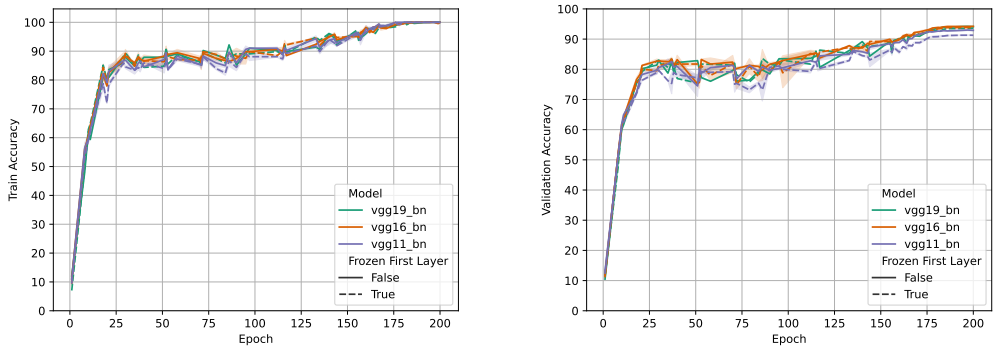
Table 3: Correlation between energy profiles of a linear CNN trained on different binary subsets of CIFAR10, when using true and random labels and comparing with a random initialization and the optimal solution.

Dataset	True Labels vs Optimal Solution	True Labels vs Init.	True Labels vs Random Labels
Bird vs Plane	-0.16 $\pm$ 0.006	0.03 $\pm$ 0.2	0.97 $\pm$ 0.003
Dog vs Frog	-0.22 $\pm$ 0.1	-0.09 $\pm$ 0.02	0.96 $\pm$ 0.01
Dog vs Cat	-0.25 $\pm$ 0.004	0.08 $\pm$ 0.24	0.95 $\pm$ 0.002
Car vs Truck	0.26 $\pm$ 0.004	-0.26 $\pm$ 0.09	0.94 $\pm$ 0.006
Boat vs Plane	-0.26 $\pm$ 0.002	-0.08 $\pm$ 0.08	0.94 $\pm$ 0.003

Appendix B Expanded Results on Similarity Between Pretrained CNNs

B.1 Accuracy of Networks trained with and without a frozen first layer

As shown in 3, networks of different depths converge to the same minimal loss value when trained with and without their first layer. To complement this we present the accuracies of these models below (fig. 9), echoing this result.



(a) Train Accuracy (b) Validation Accuracy

Figure 9: Training error and test error of VGGs of different depths on CIFAR10 (with crossentropy loss) as function of iteration with frozen first layer and without. For deep networks the performance is the same with frozen layer.

B.2 Comparison of Pretrained CNNs on CIFAR and ImageNet

To expand on the similarity between first layers of different architectures, we present correlation plots emphasizing the difference between a random initialization and the learned weights of different networks on different datasets. Presented are figures comparing pretrained models on ImageNet (figs. 10 and 11), CIFAR10 (fig. 12) CIFAR100 (fig. 13), and ResNets trained on different datasets (fig. 14). All models were downloaded through the Pytorch Model Hub.

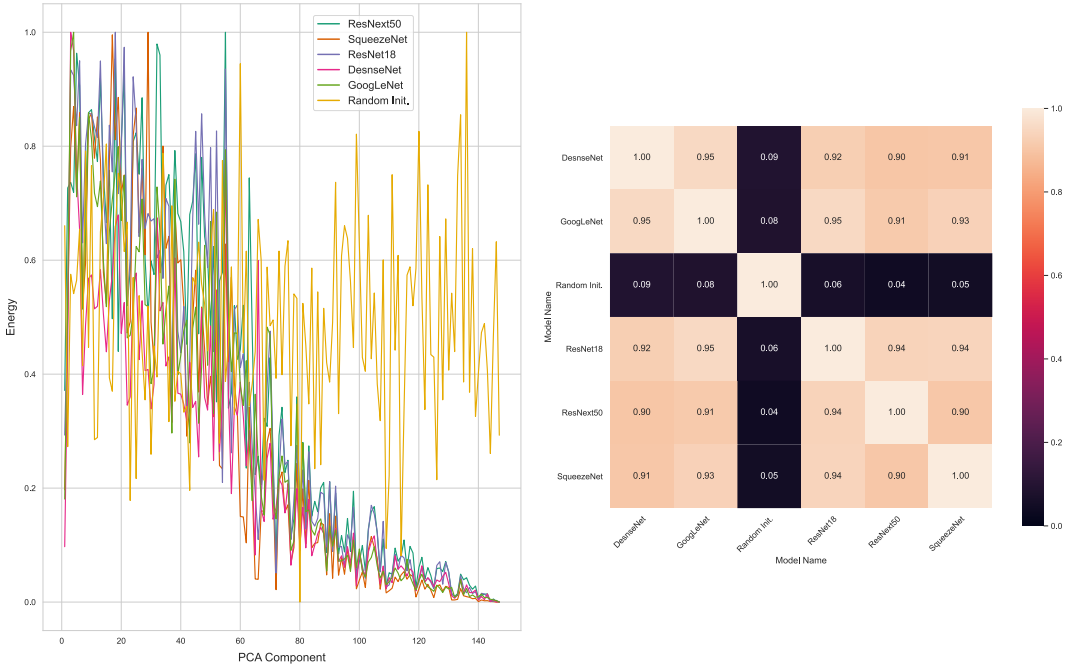


Figure 10: The energy profiles of networks with different architectures and first layer with kernel size 7, trained on Imagenet, are correlated and differ much from a random initialization.

Although it might seem odd that correlation on Imagenet is much higher than on the CIFAR datasets, we believe this is due to resolution - while on the CIFAR datasets correlation is calculated over an energy profile in $\mathbb{R}^{27}$, the Imagenet example contains profiles in $\mathbb{R}^{147}$, making the calculated

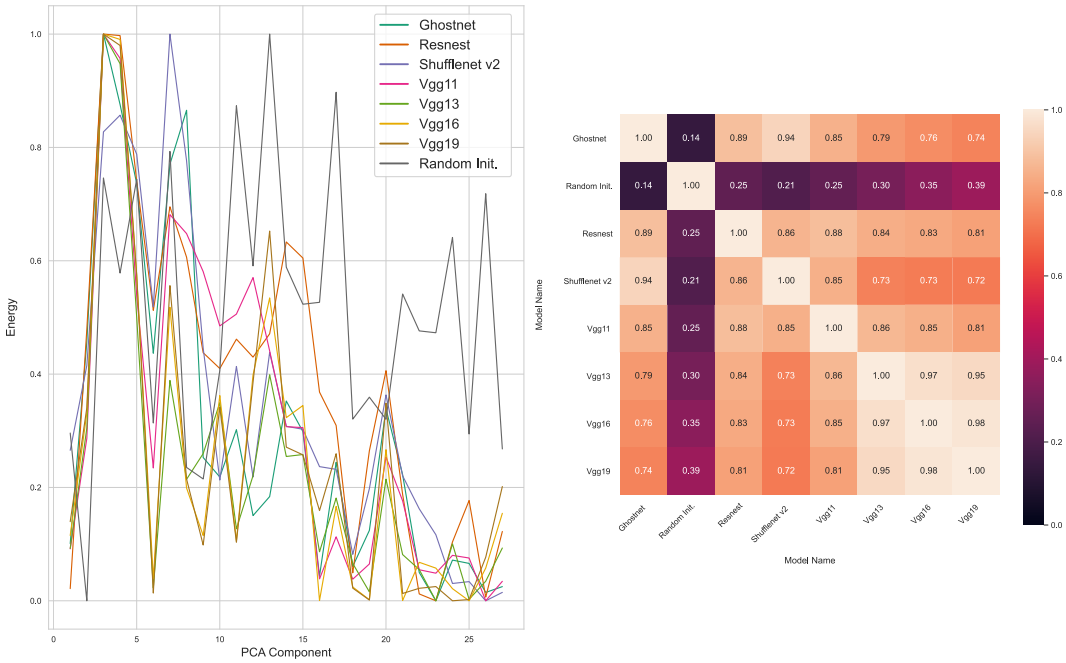


Figure 11: The energy profiles of networks with different architectures and first layer with kernel size 3, trained on Imagenet, are correlated and differ much from a random initialization.

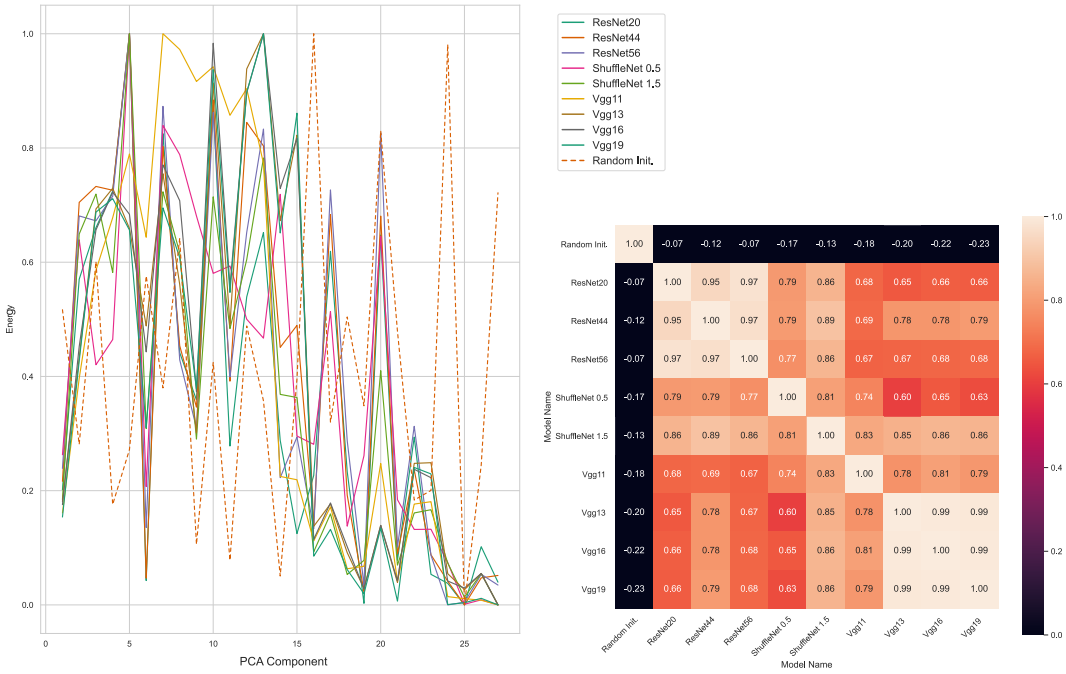


Figure 12: The energy profiles of networks with different architectures and first layer with kernel size 3, trained on CIFAR10, are correlated and differ much from a random initialization.

correlation smoother and less sensitive to noise. This is demonstrated in fig. 15 which presents the correlation between 27 components of the Imagenet profiles. When looking in higher resolution the correlation coefficients between the different models drop and are relatively equal to those between the different models on the CIFAR datasets.

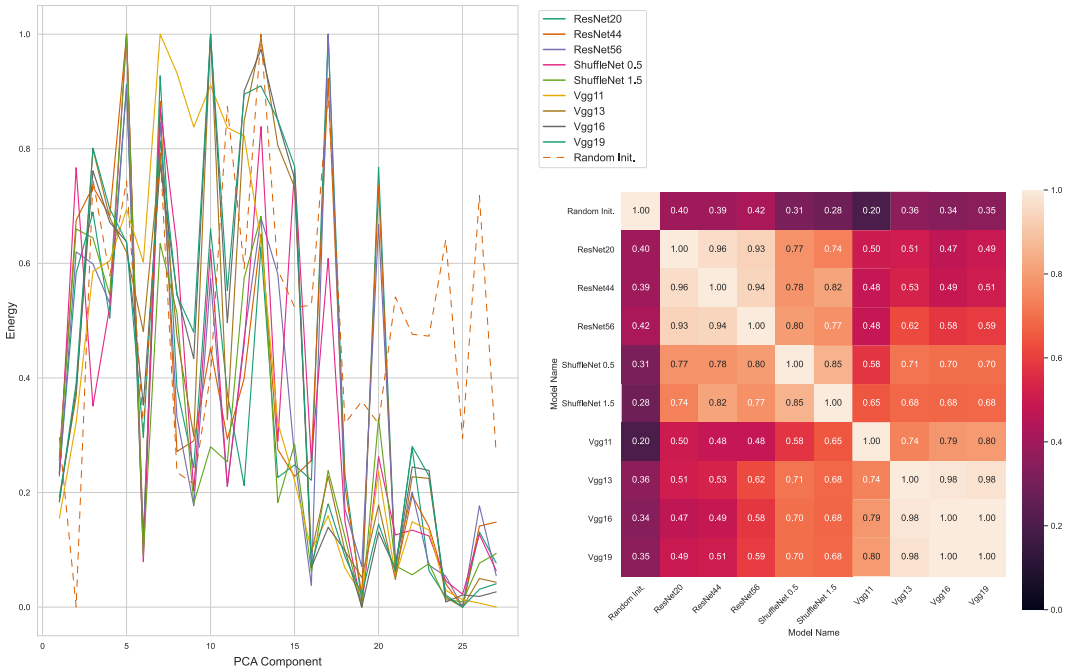


Figure 13: The energy profiles of networks with different architectures and first layer with kernel size 3, trained on CIFAR100, are correlated. Although it is possible to sample a random initialization that correlates with some models (ResNet) as good as others correlate with them (VGG 11), most models still differ from such a random initialization.

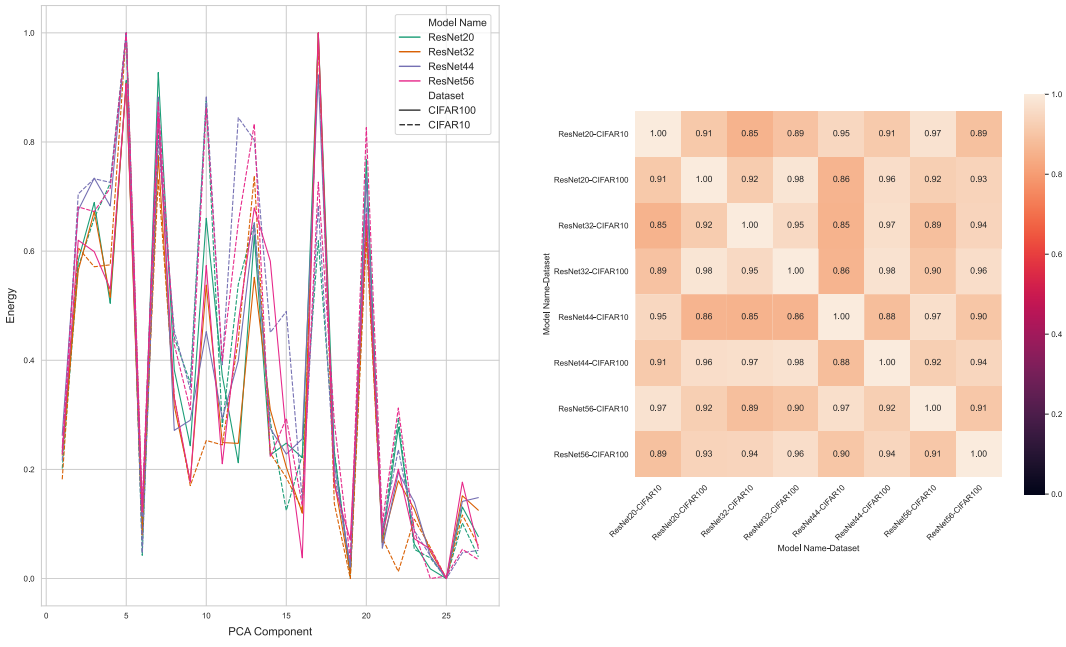


Figure 14: An expansion of the result shown in Figure 2: ResNets of different depths trained on different datasets have highly correlated energy profiles.

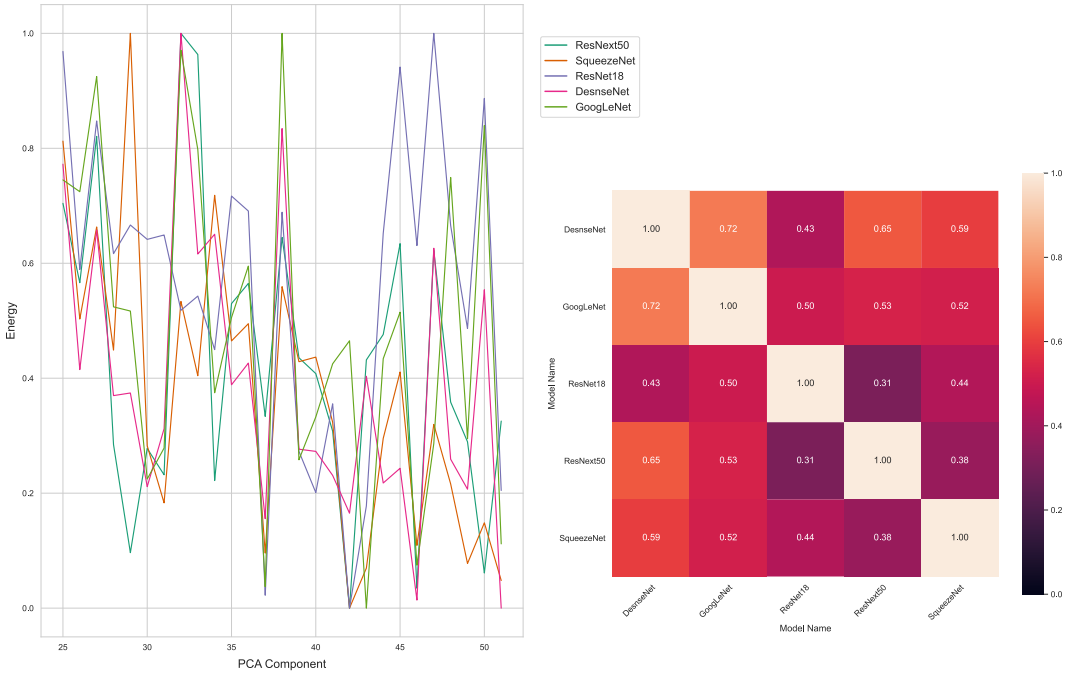


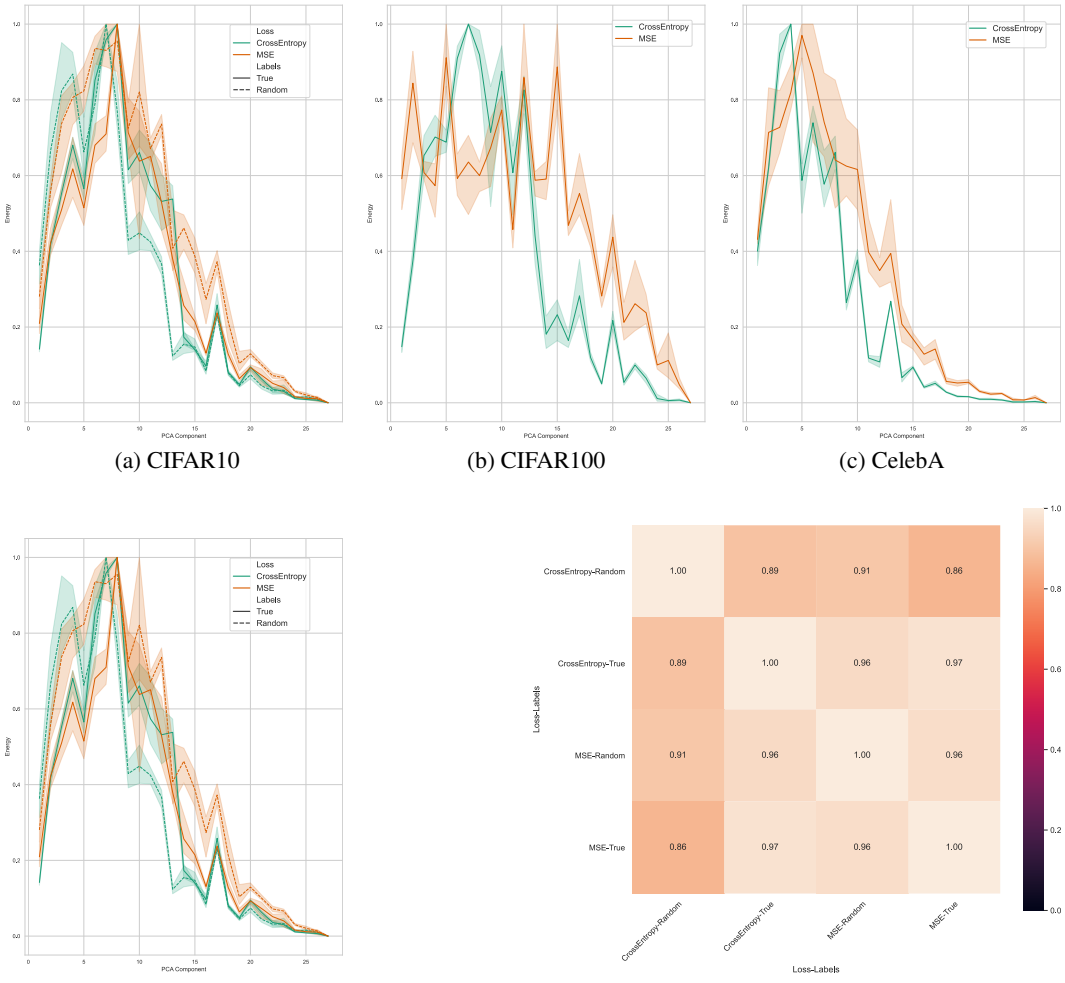
Figure 15: Correlation between different model energy profiles on Imagenet when zooming in on components 25-52. The higher correlation between the models relative to models on CIFAR is due to a higher dimension of the energy profiles.

B.3 COMPARISON OF VGG WITH DIFFERENT LOSSES

Although A.4 and all other theorems are proved on a linear network using MSE loss (as customary in theoretical works on linear networks e.g. (Hacohen & Weinshall, 2022; LeCun et al., 1991)), in practice most CNNs for multi-class classification are trained with crossentropy loss. To test the effect on the energy profile of a real network, we trained VGG with both crossentropy and MSE, and with true and random labels, the results are displayed 16 and correlations in 4. As can be seen in the figure, even in this case the networks' energy profiles are highly correlated, thus supporting our hypothesis that the main difference between the formula A.5 and the pretrained networks is due to the oversimplification of the linear model, and not for example the loss used in theory vs practice.

Table 4: Correlation between energy profiles of VGG11 when trained with MSE loss and Cross Entropy (CE) loss. Due to optimization challenges, we subtracted the initialization from the first layer prior to calculating the correlation, now comparing the accumulated gradients.

DATASET	MSE VS CROSS ENTROPY
CIFAR10	0.96 ± 0.01
CIFAR100	0.67 ± 0.03
FACES	0.90 ± 0.025



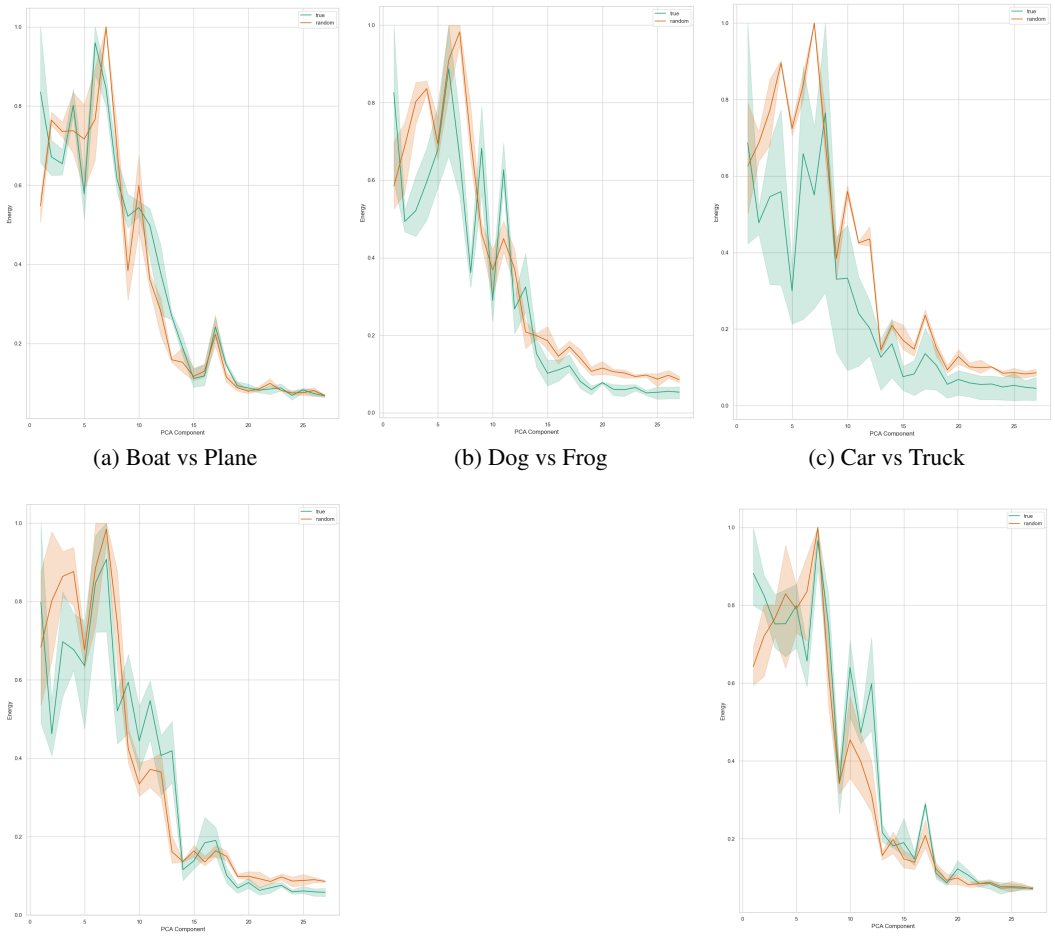
(a) CIFAR10 (b) CIFAR100 (c) CelebA

(d) Comparing Labels and Losses (CIFAR10)

(e) Correlations (CIFAR10)

Figure 16: Comparison between VGG11 trained with MSE loss and Cross Entropy loss on different datasets. Models are highly correlated, and learn similar components in their first layer. The phenomena is consistent even when training with random labels (fig. 16d). Initialization was subtracted from the first layer prior to calculation of energy profile due to the challenging optimization of networks trained with MSE.

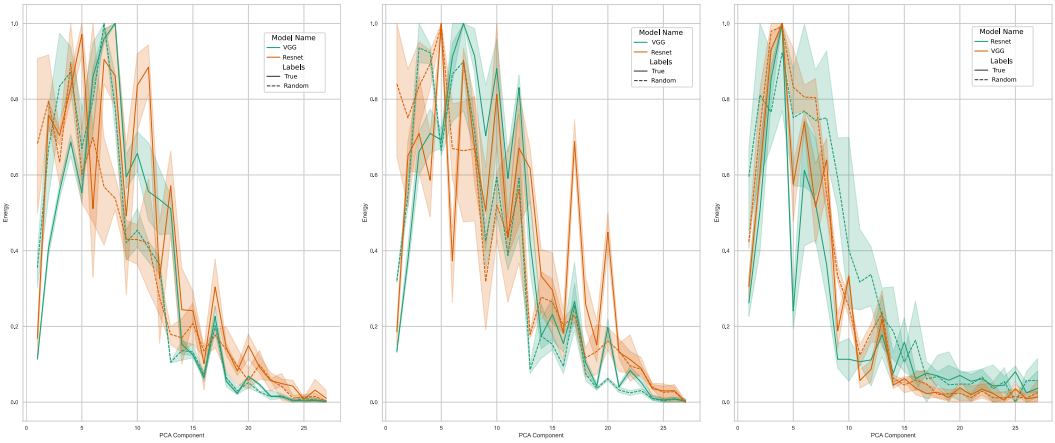
B.4 Full Figures on True and Random Labels



(a) Boat vs Plane (b) Dog vs Frog (c) Car vs Truck

(d) Dog vs Cat (e) Bird vs Plane

Figure 17: VGG11 trained on binary classification tasks from CIFAR10 exhibit similar energy patterns when trained with true and random labels.



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

Figure 18: CNNs trained on a CIFAR10, CIFAR100 and a CelebA classification task exhibit similar energy patterns when trained with true and random labels, when using both VGG and ResNet architectures.

B.5 EXPERIMENTAL DETAILS

All models - linear and non linear were trained with SGD and a constant learning rate of 0.1. No preprocessing was applied to the data except when stated otherwise. All models were trained for 150 epochs, with minibatches of size 256. All results are averaged over at least 3 different random seeds.
When referring to models "trained with random labels", we trained models until they overfit the training data, as both ResNet and VGG can reach 99% train accuracy on CIFAR10 with random labels.

All models in the main text were trained ourselves, except those depicted in 1. All pretrained models in 1 and B.2 were downloaded from the Pytorch Model Hub.