DO WIDE AND DEEP NETWORKS LEARN THE SAME THINGS? UNCOVERING HOW NEURAL NETWORK REPRESENTATIONS VARY WITH WIDTH AND DEPTH

Thao Nguyen,* Maithra Raghu, & Simon Kornblith
Google Research
{thaotn,maithra,skornblith}@google.com

김 성 철

Contents

Introduction

112	Probabilistic Numeric Convolutional Neural Networks	https://open	6.75	['7', '6', '7', '7']
113	Deep Representational Re-tuning using Contrastive Tension	https://open	6.75	['6', '7', '5', '9']
114	Multiplicative Filter Networks	https://open	6.75	['6', '6', '6', '9']
115	Improved Autoregressive Modeling with Distribution Smoothing	https://open	6.75	['8', '6', '7', '6']
116	Gradient Vaccine: Investigating and Improving Multi-task Optimization in M	https://open	6.75	['6', '6', '8', '7']
117	Do Wide and Deep Networks Learn the Same Things? Uncovering How Ne	https://open	6.75	['7', '6', '8', '6']
118	Negative Data Augmentation	https://open	6.75	['5', '6', '9', '7']
119	Molecule Optimization by Explainable Evolution	https://open	6.75	['7', '6', '6', '8']
120	Structured Prediction as Translation between Augmented Natural Languag	https://open	6.75	['7', '6', '8', '6']
121	A Better Alternative to Error Feedback for Communication-Efficient Distrib	https://open	6.75	['5', '6', '7', '9']
122	Randomized Automatic Differentiation	https://open	6.75	['4', '8', '8', '7']

Introduction

- **모델의 width(channel)와 depth의 선정에는 많은 선택의 폭이 존재**

  → 최종 모델에 어떤 영향을 미치는지에 대한 이해는 제한적

  - Depth와 width가 최종 학습된 representation에 어떤 영향을 미칠까?
  - 각기 다른 모델들은 다른 intermediate feature를 학습할까?
  - 결과에 대해 눈에 보이는 차이가 존재할까?

- ResNet family를 다양한 depth와 width에서 실험
  - CIFAR-10, CIFAR-100, and ImageNet
  - 모델 내부의 representations & 각기 다른 모델의 초기값 및 구조에 따른 결과

Introduction

- **Contributions**
  - CKA (Centered Kernel Alignment)를 통해 다른 neural network architecture의 hidden representation을 측정
    - *Block structure* : wide or deep model의 representation에서 나타나는 특징 구조
  - Block structure는 **single principle component**를 갖는 hidden representation과 일치함을 발견
    - Single principle component : 해당 레이어를 통해 보존되고 전파되는 representation의 variance의 대부분을 설명
  - 다른 구조의 representation들을 비교
  - Depth와 width가 모델의 결과에 얼마나 다른 영향을 미치는지 확인

Experimental Setup and Background

- **Representational Similarity Measures**
  - Linear Centered Kernel Alignment (CKA)
    - Centering matrix : symmetric and idempotent matrix. 벡터에 곱해줄 경우, 벡터의 평균을 빼준 것과 같은 효과

Definition [edit]

The **centering matrix** of size n is defined as the n-by-n matrix

$$C_n = I_n - \tfrac{1}{n}\mathbb{O}$$

where I_n is the identity matrix of size n and $\mathbb{O}$ is an n-by-n matrix of all 1's. This can also be written as:

$$C_n = I_n - \tfrac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$$

where $\mathbf{1}$ is the column-vector of n ones and where $^{\mathrm{T}}$ denotes matrix transpose.

For example

$$C_1 = [0],$$

$$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

- *(2) Prerequisite : Centering Matrix*
  - » $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{n\times 1}$, *and* $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^T$

    where $\mathbf{I}_n$ is an identity matrix of size n (n = *#objects*)
  - » Based the centering matrix H, given a vector **x**

    $$\mathbf{H}\mathbf{x} = \mathbf{x} - \left(\frac{1}{n}\mathbf{1}^T\mathbf{x}\right)\mathbf{1}$$

    mean of vector
  - » Its use is to make easier matrix manipulations

https://en.wikipedia.org/wiki/Centering_matrix
http://fourier.eng.hmc.edu/e176/lectures/algebra/node13.html

Experimental Setup and Background

- **Representational Similarity Measures**
  - Linear Centered Kernel Alignment (CKA)
    - Let $\mathbf{X} \in \mathbb{R}^{m \times p_1}$ and $\mathbf{Y} \in \mathbb{R}^{m \times p_2}$ contain activations of two layers, one with p_1 neurons and another p_2 neurons
    - The elements of the Gram matrices $\mathbf{K} = \mathbf{X}\mathbf{X}^\mathbf{T}$ and $\mathbf{L} = \mathbf{Y}\mathbf{Y}^\mathbf{T}$ reflect the similarities between pairs of examples
    - Let $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\mathbf{T}$ be the centering matrix
    - $\mathbf{K}' = \mathbf{HKH}$ and $\mathbf{L}' = \mathbf{HLH}$ reflect the similarity matrices with their column and row means subtracted
    - **HSIC (Hilbert–Schmidt Independence Criterion)** measures the similarity of these reshaped similarity matrices by reshaping them to vectors and taking the dot product between these vectors, $\mathrm{HSIC}_0(\mathbf{K}, \mathbf{L}) = \mathrm{vec}(\mathbf{K}') \cdot \mathrm{vec}(\boldsymbol{L}')/(m-1)^2$
    - HSIC is invariant to orthogonal transformations of the representations and, by extension, to permutation of neurons, but it is not invariant to isotropic scaling of the original representations
    - CKA further normalizes HSIC to produce a similarity index between 0 and 1 that is invariant to isotropic scaling,

$$\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}_0(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}_0(\mathbf{K}, \mathbf{K})\mathrm{HSIC}_0(\mathbf{L}, \mathbf{L})}}$$

https://openreview.net/forum?id=KJNcAkY8tY4

Experimental Setup and Background

- **Representational Similarity Measures**
  - Linear Centered Kernel Alignment (CKA)
    - 메모리소모를 줄이기 위해, CKA를 k minibatch에 대해 HSIC의 평균으로 사용

$$\mathrm{CKA} = \frac{\frac{1}{\mathrm{k}}\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{HSIC}_1\left(\mathrm{X}_i\mathrm{X}_i^{\mathrm{T}}, \mathrm{Y}_i\mathrm{Y}_i^{\mathrm{T}}\right)}{\sqrt{\frac{1}{k}\sum_{i=1}^{k} \mathrm{HSIC}_1\left(\mathrm{X}_i\mathrm{X}_i^{\mathrm{T}}, \mathrm{X}_i\mathrm{X}_i^{\mathrm{T}}\right)}\sqrt{\frac{1}{k}\sum_{i=1}^{k} \mathrm{HSIC}_1\left(\mathrm{Y}_i\mathrm{Y}_i^{\mathrm{T}}, \mathrm{Y}_i\mathrm{Y}_i^{\mathrm{T}}\right)}}$$

    - HSIC의 unbiased estimator를 사용하여 CKA의 값이 batch size에 독립적이도록 함

$$\mathrm{HSIC}_1(\mathbf{K}, \mathbf{L}) = \frac{1}{n(n-3)}\left(\mathrm{tr}\left(\tilde{\mathbf{K}}\tilde{\mathbf{L}}\right) + \frac{1^{\mathrm{T}}\tilde{\mathbf{K}}1 1^{\mathrm{T}}\tilde{\mathbf{L}}1}{(n-1)(n-2)} - \frac{2}{n-2}1^{\mathrm{T}}\tilde{\mathbf{K}}\tilde{\mathbf{L}}1\right)$$

      - $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{L}}$ are obtained by setting the diagonal entries of $\mathbf{K}$ and $\mathbf{L}$ to zero

https://openreview.net/forum?id=KJNcAkY8tY4

Depth, Width and Model Internal Representations

- **Main questions**
  - How do representations evolve through the hidden layers in different architectures?
  - How similar are different hidden layer representations to each other?

- **Internal Representations and the Block Structure**
  - CIFAR-10에서 다양한 depth와 width의 ResNet 결과
    - CKA를 사용해서 각 모델 내 레이어의 모든 pair의 representation similarity를 계산

Depth, Width and Model Internal Representations

- **Internal Representations and the Block Structure**
  - CIFAR-10에서 다양한 depth와 width의 ResNet 결과
    - CKA를 사용해서 모든 layer pair의 representation similarity를 계산
    - 모델이 넓어지거나 깊어질수록, **block structure** 발생
      - 아주 높은 representation similarity를 갖는 hidden layer의 큰 범위
    - Block structure는 뒤쪽 layer (마지막 2stage)에 자주 나타남
      - Residual connection이 없는 경우도 유사한 결과 확인
    - 각기 다른 seed에서도 CKA확인
      - Seed마다 크기와 위치가 달라도 block structure는 확실히 보임

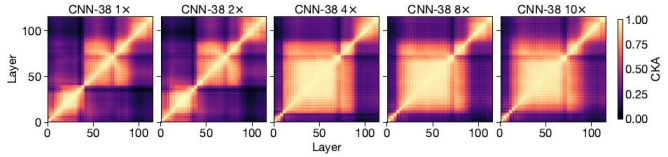


Figure C.1: Block structure also appears in models without residual connections. We remove residual connections from existing CIFAR-10 ResNets and plot CKA heatmaps for layers in the resulting architecture after training. Since the lack of residual connections prevents deep networks from performing well on the task, here we only show the representational similarity for models of increasing width. As previously observed in Figure 1, the block structure emerges in higher capacity models.

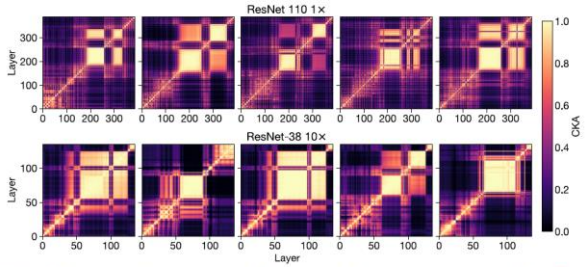


Figure D.1: Block structure varies across random initializations. We plot CKA heatmaps as in Figure 1 for 5 random seeds of a deep model (top row) and a wide model (bottom row) trained on CIFAR-10. While the size and position vary, the block structure is clearly visible in all seeds.

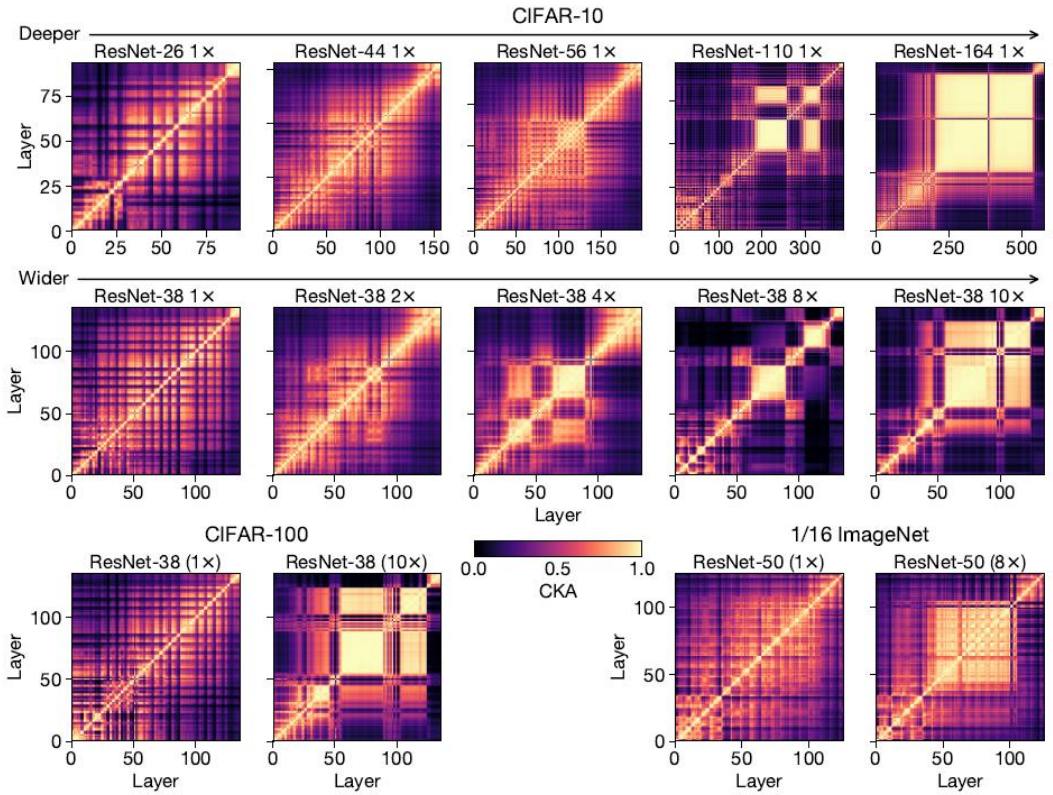


Figure 1: Emergence of the *block structure* with increasing width or depth. As we increase the depth or width of neural networks, we see the emergence of a large, contiguous set of layers with very similar representations — the block structure. Each of the panes of the figure computes the CKA similarity between all pairs of layers in a single neural network and plots this as a heatmap, with x and y axes indexing layers. See Appendix Figure C.1 for block structure in wide networks without residual connections.

Depth, Width and Model Internal Representations

- **The Block Structure and Model Overparametrization**
  - Block structure가 절대적인 모델 크기 또는 training data의 크기와 비교했을 때 모델 크기와 관련이 있는지 확인
    - 모델구조를 고정하고 training dataset 크기를 줄여 상대적으로 model capacity를 확장
    - 적은 training data가 사용되었을 때 narrower or shallower (lower capacity) network에서도 block structure가 발견됨

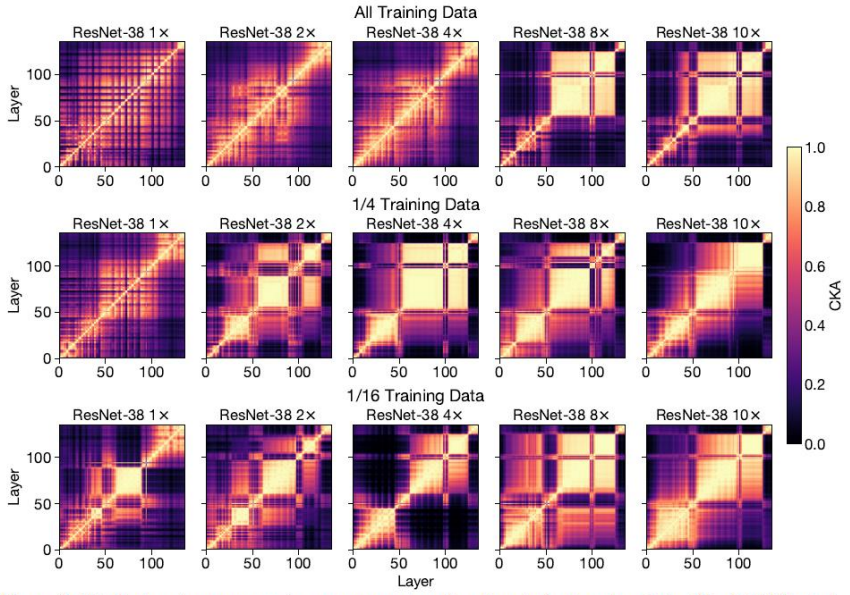


Figure 2: Block structure emerges in narrower networks when trained on less data. We plot CKA similarity heatmaps as we increase network width (going right along each row) and also decrease the dataset size (down each column). As a result of the increased model capacity (with respect to the task) from smaller dataset size, smaller (narrower) models now also exhibit the block structure.

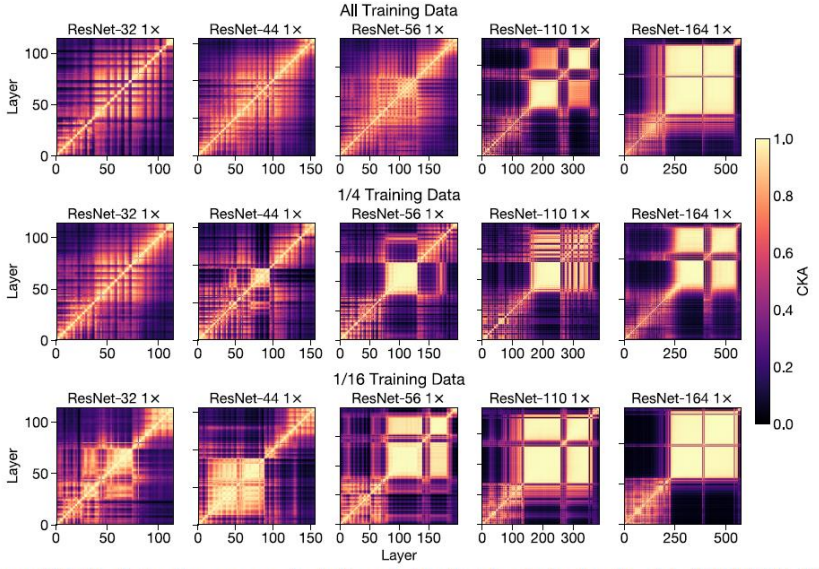


Figure D.2: Block structure emerges in shallower networks when trained on less data (CIFAR-10). We plot CKA similarity heatmaps as we increase network depth (going right along each row) and also decrease the size (down each column) of training data. Similar to the observation made in Figure 2, as a result of the increased model capacity (with respect to the task) from smaller dataset size, smaller (shallower) models now also exhibit the block structure.

Probing the Block Structure

- **Main questions**
  - Block structure는 wide and/or deep neural network와 task에 비해 model capacity가 큰 경우에 발생
    - What is happening to the neural network representations as they propagate through the block structure?
- **The Block Structure and the First Principal Component**
  - Linear CKA for centered matrices of activations $\mathbf{X} \in \mathbb{R}^{n \times p_1}$, $\mathbf{Y} \in \mathbb{R}^{n \times p_2}$

$$\mathrm{CKA}\left(XX^{\mathrm{T}}, YY^{\mathrm{T}}\right) = \frac{\sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \lambda_X^k \lambda_Y^j \left\langle \mathbf{u}_X^i, \mathbf{u}_X^j \right\rangle^2}{\sqrt{\sum_{i=1}^{p_1} \left(\lambda_X^i\right)^2} \sqrt{\sum_{j=1}^{p_2} \left(\lambda_Y^j\right)^2}}$$

  - $\mathbf{u}_X^i \in \mathbb{R}^n$, $\mathbf{u}_Y^i \in \mathbb{R}^n$: the i^{th} normalized principal components of $\mathbf{X}$ and $\mathbf{Y}$
  - λ_X^i, λ_Y^i : the corresponding squared singular values
  - Eq.3이 1에 가까워질 때, CKA는 $\langle \mathbf{u}_X^1, \mathbf{u}_Y^1 \rangle^2$사이의 squared alignment에 반영

Probing the Block Structure

- **The Block Structure and the First Principal Component**
  - Block structure가 존재하는 network에서 first principal component가 분산의 큰 부분을 나타내고, block structure가 없는 경우는 그렇지 않음

    → block structure가 representation의 first principal component의 행동을 반영함을 나타냄

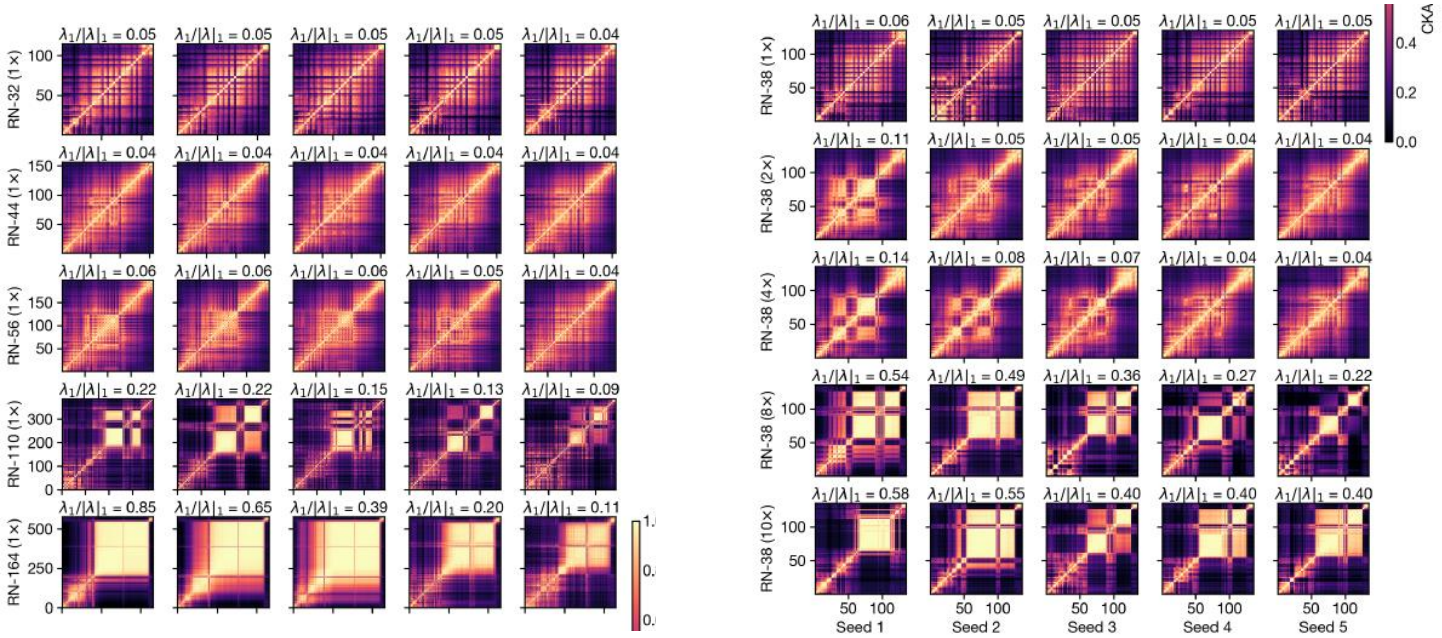
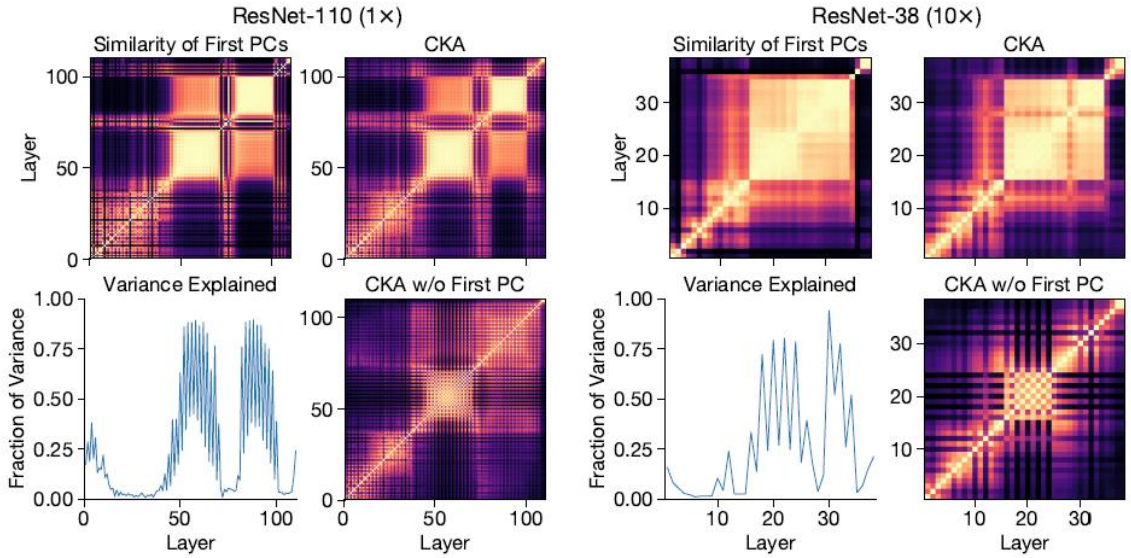


Figure D.5: Top principal component explains a large fraction of variance in the activations of models with block structure. Each row shows a different model configuration that is trained on CIFAR-10, with the first 5 rows showing models of increasing depth, and the last 5 rows models of increasing width. Columns correspond to different seeds. Each heatmap is labeled with the fraction of variance explained by the top principal component of activations combined from the last 2 stages of the model (where block structure is often found). Rows (seeds belonging to the same architecture) are sorted by decreasing value of the proportion of variance explained. We observe that this variance measure is significantly higher in model seeds where the block structure is present.

Probing the Block Structure

- **The Block Structure and the First Principal Component**
  - Block structure와 first principal component 사이의 관계
    - First principal component (bottom left)와 location of the block structure (top right)
      → block structure에 속한 layer들은 매우 큰 first principal component를 가지고 있음
    - Cosine similarity of the first principal components across all pairs of layers (top left)
      → block structure와 비슷한 구조
    - First principal component를 제거 (bottom right)
      → CKA heatmap의 block structure가 거의 없어짐

→ block structure는 구성 layer들에 걸쳐 first principal component를 preserving & propagating하며 발생!



Probing the Block Structure

- **Linear Probes and Collapsing the Block Structure**
  - Block structure가 representation의 key component를 보존한다는 점에서
    - Network의 task performance에 영향을 미치는지
    - 성능에 최소한의 영향으로 block structure를 없앨 수 있는지
  - Network의 각 layer에서 linear probe를 학습

3.2 Linear classifier probes

Consider the common scenario in deep learning in which we are trying to classify the input data X to produce an output distribution over D classes. The last layer of the model is a densely-connected map to D values followed by a softmax, and we train by minimizing cross-entropy.

At every layer we can take the features H_k from that layer and try to predict the correct labels y using a linear classifier parameterized as

$$f_k : H_k \to [0,1]^D$$
$$h_k \mapsto \text{softmax}\left(W h_k + b\right).$$

where $h_k \in H$ are the features of hidden layer k, $[0,1]^D$ is the space of categorical distributions of the D target classes, and (W, b) are the probe weights and biases to be learned so as to minimize the usual cross-entropy loss.

Let $\mathcal{L}_k^{\text{train}}$ be the empirical loss of that linear classifier f_k evaluated over the training set. We can also define $\mathcal{L}_k^{\text{valid}}$ and $\mathcal{L}_k^{\text{test}}$ by exporting the same linear classifier on the validation and test sets.

Without making any assumptions about the model itself being trained, we can nevertheless assume that these f_k are themselves optimized so that, at any given time, they reflect the currently optimal thing that can be done with the features present.

We refer to those linear classifiers as "probes" in an effort to clarify our thinking about the model. These probes do not affect the model training. They only measure the level of linear separability of the features at a given layer. Blocking the backpropagation from the probes to the model itself can be achieved by using `tf.stop_gradient` in Tensorflow (or its Theano equivalent), or by managing the probe parameters separately from the model parameters.

https://arxiv.org/pdf/1610.01644.pdf

Probing the Block Structure

- **Linear Probes and Collapsing the Block Structure**
  - Network의 각 layer에서 linear probe를 학습
    - Block structure가 없는 경우 → monotonic increase in accuracy
    - Block structure가 있는 경우 → little improvement inside the block structure
    - The accuracies of probes for pre- and post-residual connections
      → residual connection들이 block structure에서 representation을 보존하는데 중요한 역할

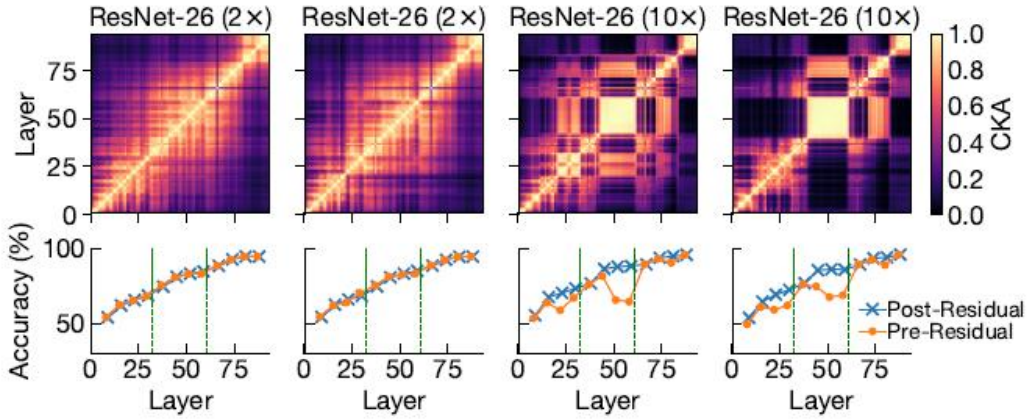


Figure 4: Linear probe accuracy. Top: CKA between layers of individual ResNet models, for different architectures and initializations. Bottom: Accuracy of linear probes for each of the layers before (orange) and after (blue) the residual connections.

Probing the Block Structure

- **Linear Probes and Collapsing the Block Structure**
  - Residual connection을 유지하면서 각 residual stage의 끝부터 block을 하나씩 pruning
    - Middle stage에서 block들이 drop되었을 때 test accuracy에 작은 영향을 미침 (with block structure)
    - 다른 seed들로 비교했을 때, 정확도의 하락의 크기는 block structure의 크기와 선명도와 관련이 있음

    → Block structure는 model design관점에서 불필요한 모듈의 indicator

    → Constituent layer representation의 similarity는 model compression에 이용될 수 있음

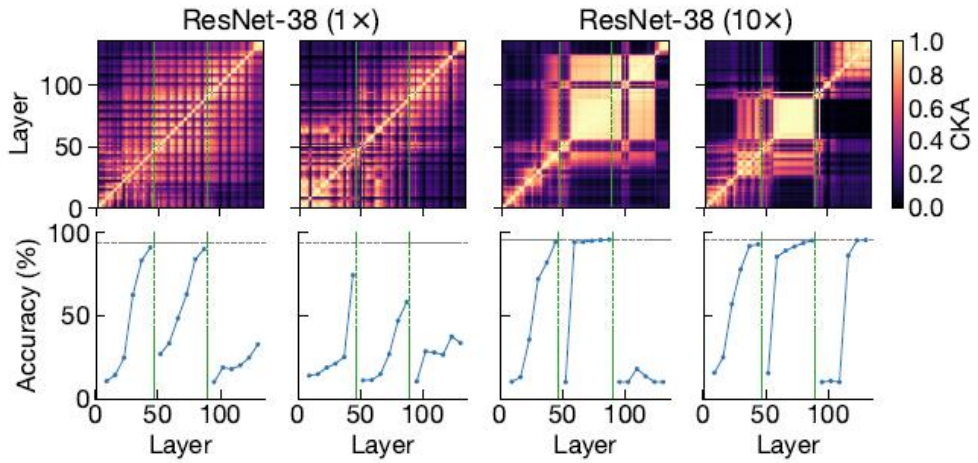


Figure 5: Effect of deleting blocks on accuracy for models with and without block structure. Blue lines show the effect of deleting blocks backwards one-by-one within each ResNet stage. (Note the plateau at the block structure.) Vertical green lines reflect boundaries between ResNet stages. Horizontal gray line reflects accuracy of full model.

Depth and Width Effects on Representations across Models

- **Main questions**
  - How depth and width affect the hidden representations across models?
    - Are learned representations similar across models of different architectures and different random initializations?
    - How is this affected as model capacity is changed?

Depth and Width Effects on Representations across Models

- **같은 모델로 다른 training runs 사이의 representation의 변화**
  - CKA heatmaps for a smaller model (left), wide model (middle), and deep model (right)
    - Smaller model (ResNet-38 (1x))은 block structure 없음

      → 모든 seed에 대해 representation들이 유사한 구조를 보임
    - Wide and deep model은 block structure가 존재 (diagonal plots)

      → block structure에 속하지 않는 layer들은 유사성을 보이는데, block structure representation은 매우 유사하지 않음

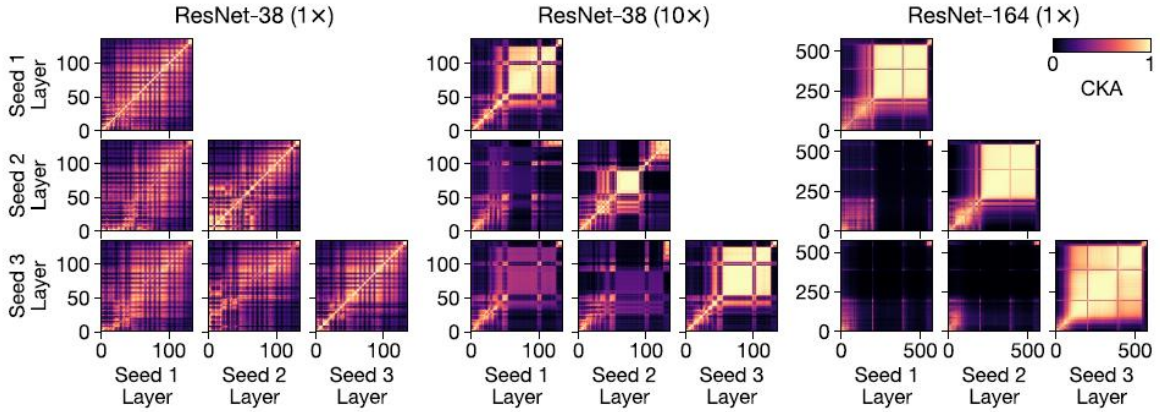


Figure 6: Representations within "block structure" differ across initializations. Each group of plots shows CKA between layers of models with the same architecture but different initializations (off the diagonal) or within a single model (on the diagonal). For narrow, shallow models such as ResNet-38 (1×), there is no block structure, and CKA across initializations closely resembles CKA within a single model. For wider (middle) and deeper (right) models, representations within the block structure are highly dissimilar.

Depth and Width Effects on Representations across Models

- **다른 모델들의 CKA비교**
  - Wide and deep model without the block structure

    → representation similarity를 보이고, 해당하는 레이어들은 대체로 same proportional depth
  - With block structure

    → block structure representation은 각 모델에 대해 고유하게 유지

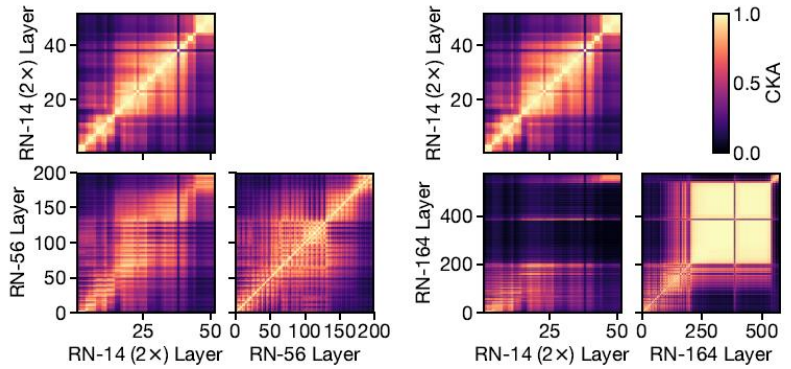


Figure E.1: Representations align between models of different widths and depths when no block structure is present. In each group of heatmaps, top left and bottom right show CKA within a single model trained on CIFAR-10. Bottom left shows CKA for all pairs of layers between these (non-architecturally-identical) models, which have similar test performance. In the absence of block structure (left group), representations at the same relative depths are similar across models. But when comparison involves models with block structure (right group), representations within the block structure are dissimilar to those of the other model.

Depth, Width and Effects on Model Predictions

- **Main questions**
  - How the characteristic properties of internal representations influence the outputs of the model?
    - How diverse are the predictions of different architectures?
    - Are there examples that wide networks are more likely to do well on compared to deep networks, and vice versa?
- **CIFAR-10과 ImageNet에서 network 모집단 학습**
  - Individual example level에서 output prediction의 상당한 다양성 존재
  - 유사한 구조의 architecture들은 유사한 output prediction을 가지고 있음
  - (ImageNet) wide and deep model들 사이에는 class-level error의 유의한 차이가 존재,
    특히 wide model의 경우 object보다 scene에 해당하는 클래스를 식별하는데 작은 이점을 보임

Depth, Width and Effects on Model Predictions

- CIFAR-10과 ImageNet에서 network 모집단 학습
  - 구조적으로 동일한 deep model (ResNet-62)와 wide model (ResNet-14 (2x)) 각각 100개의 accuracy 비교
    - Average accuracy가 통계적으로 구별될 수 없지만, 다른 오류를 만드는 경향이 있고, 집단간 차이는 생각보다 큼

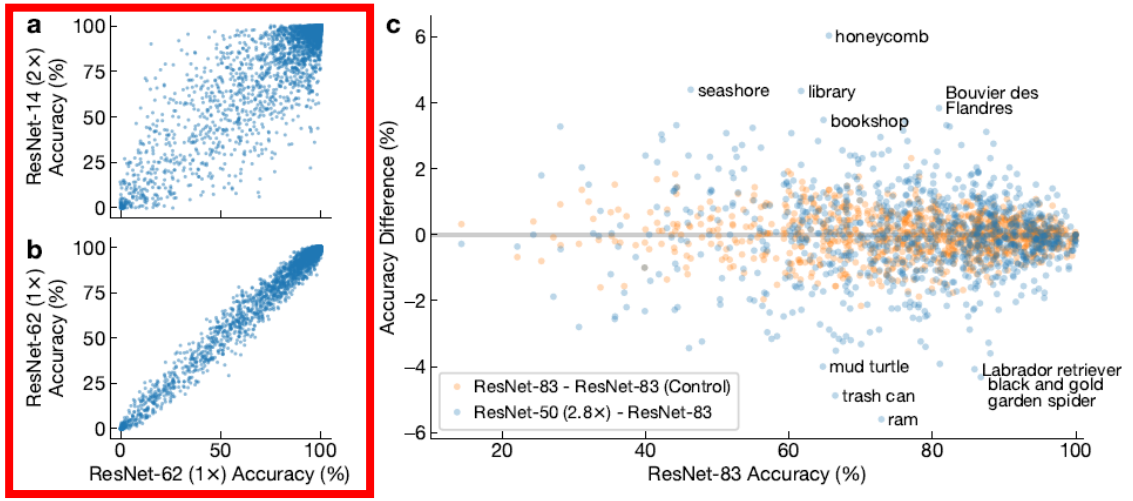


Figure 7: Systematic per-example and per-class performance differences between wide and deep models. **a**: Comparison of accuracy on individual examples for 100 ResNet-62 (1×) and ResNet-14 (2×) models, which have statistically indistinguishable accuracy on the CIFAR-10 test set. **b**: Same as (a), for disjoint sets of 100 architecturally identical ResNet-62 models trained from different initializations. See Figure F.1 for a similar plot for ResNet-14 (2×) models. **c**: Accuracy differences on ImageNet classes for ResNets between models with increased width (y-axis) or depth (x-axis) in the third stage. Orange dots reflect difference between two sets of 50 architecturally identical deep models (i.e., different random initializations of ResNet-83).

Depth, Width and Effects on Model Predictions

- CIFAR-10과 ImageNet에서 network 모집단 학습
  - 구조적으로 동일한 deep model (ResNet-62)와 wide model (ResNet-14 (2x)) 각각 100개의 accuracy 비교
    - Average accuracy가 통계적으로 구별될 수 없지만, 다른 오류를 만드는 경향이 있고, 집단간 차이는 생각보다 큼

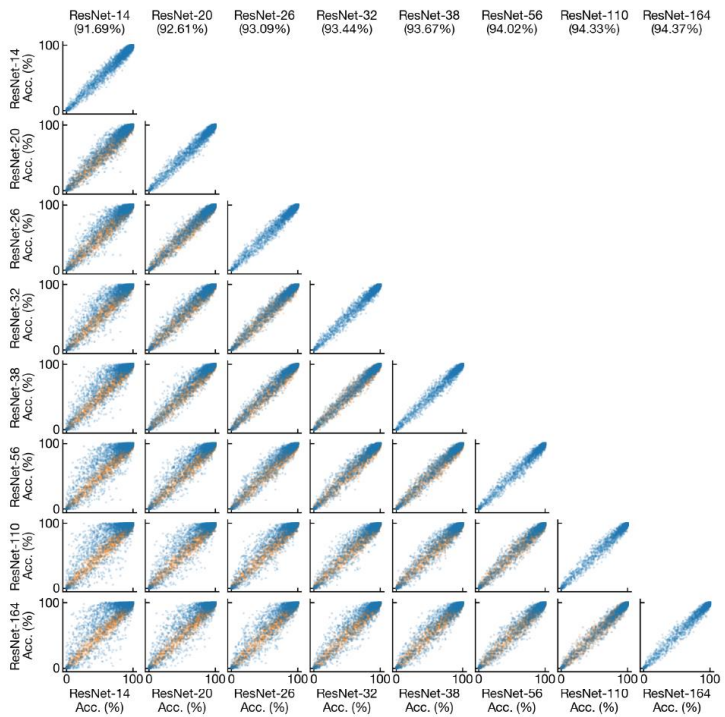


Figure F.2: Effect of depth on example accuracy. Scatter plots of per-example accuracies of ResNet models with different depths on CIFAR-10. Blue dots indicate per-example accuracies of two groups of 100 networks each with different architectures indicated by axes labels. Orange dots show the distribution for groups of architecturally identical models, copied from the plot on the diagonal above. Accuracy of each model is shown at the top.

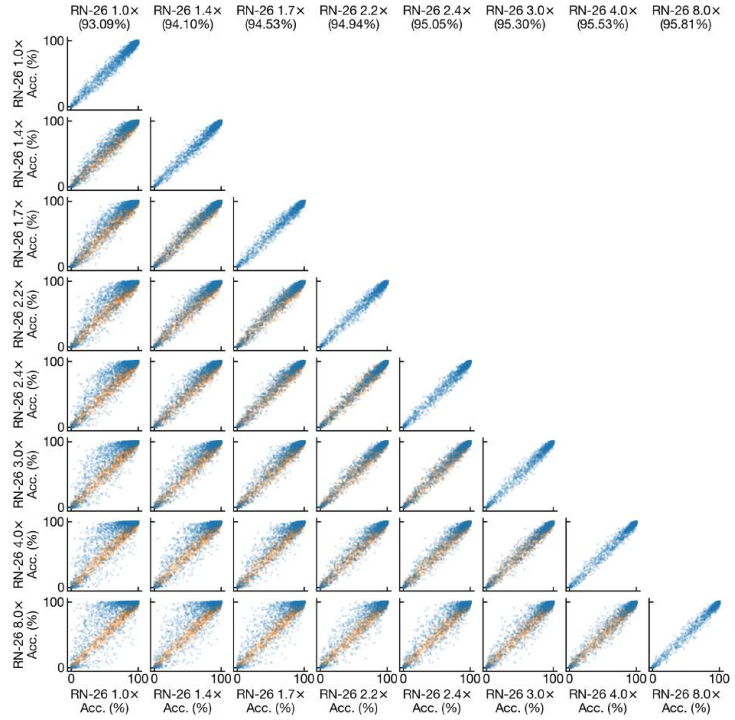


Figure F.3: Effect of width on example accuracy. Scatter plots of per-example accuracies of ResNet models with different widths on CIFAR-10. Blue dots indicate per-example accuracies of two groups of 100 networks each with different architectures indicated by axes labels. Orange dots show the expected distribution for groups of architecturally identical models, copied from the plot on the diagonal above. Accuracy of each model is shown at the top.

Depth, Width and Effects on Model Predictions

- **CIFAR-10과 ImageNet에서 network 모집단 학습**
  - Architecture가 wider or deeper로 될수록, accuracy는 증가하고, 그 효과는 smaller network에서 주로 나타남

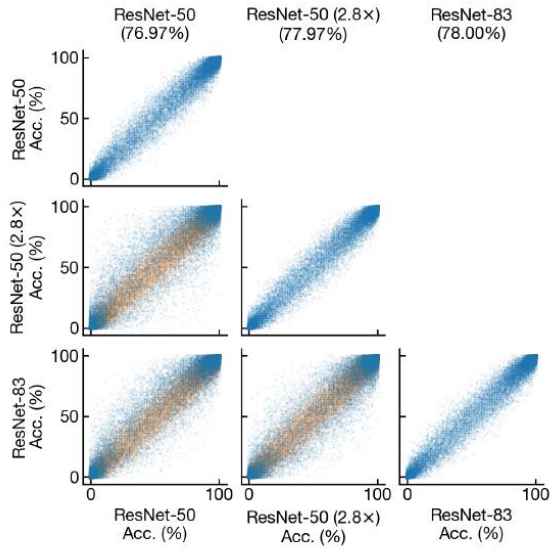


Figure F.4: Systematic per-example performance differences between wide and deep models on ImageNet. Scatter plots of per-example accuracy averaged across 50 vanilla ResNet-50 (1×) models versus that for groups of 50 models with increased depth (6 → 17 blocks, "ResNet-83") or width (2.8× wider) in the 3rd stage. Orange dots in plots show the expected distribution for two groups of 50 architecturally identical models, copied from the plot on the diagonal above. The deeper and wider models have very similar but statistically distinguishable accuracy (mean ± SEM for deeper model: 78.00 ± 0.01, wider model: 77.97 ± 0.01, $t(99) = 2.0, p = 0.047$).

Depth, Width and Effects on Model Predictions

- **Wide and deep ImageNet model이 class level에서 체계적인 정확도 차이를 보일까?**
  - 419/1000 클래스에 대한 accuracy의 통계적으로 유의미한 차이 ($p < 0.05$, Welch's t-test)
    - Wide model이 잘 분류한 top 5 클래스는 object보다 scene (seashore, library, bookshop, …)

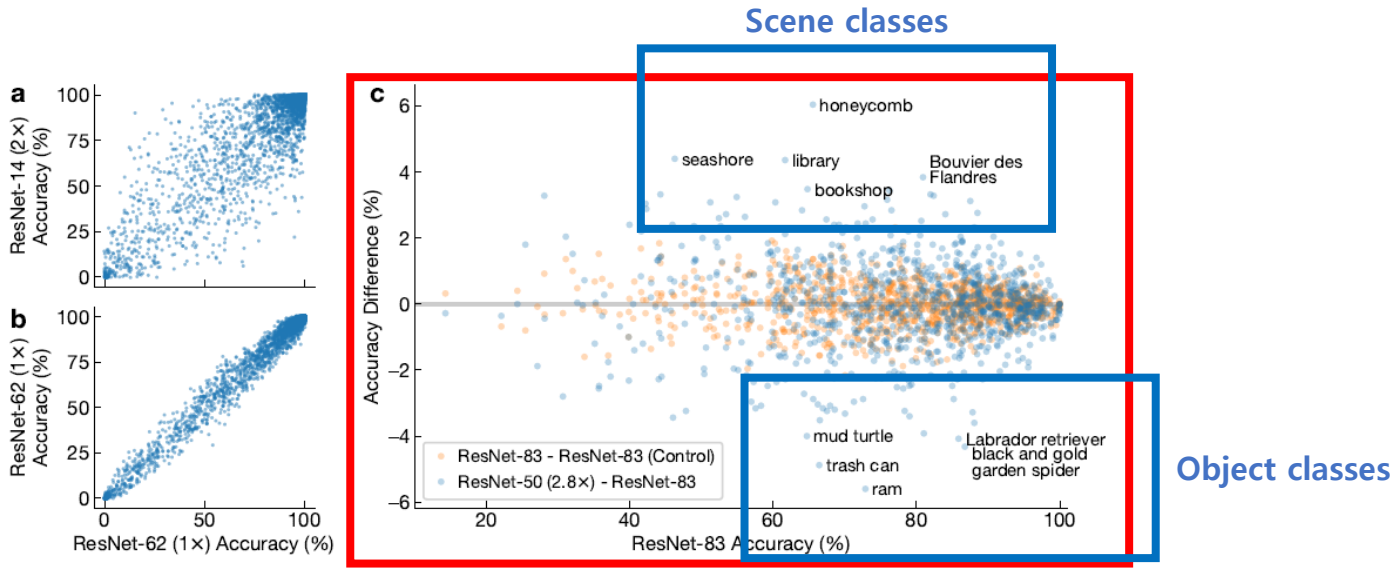


Figure 7: Systematic per-example and per-class performance differences between wide and deep models. **a**: Comparison of accuracy on individual examples for 100 ResNet-62 (1×) and ResNet-14 (2×) models, which have statistically indistinguishable accuracy on the CIFAR-10 test set. **b**: Same as (a), for disjoint sets of 100 architecturally identical ResNet-62 models trained from different initializations. See Figure F.1 for a similar plot for ResNet-14 (2×) models. **c**: Accuracy differences on ImageNet classes for ResNets between models with increased width (y-axis) or depth (x-axis) in the third stage. Orange dots reflect difference between two sets of 50 architecturally identical deep models (i.e., different random initializations of ResNet-83).

Conclusion

- Neural network representation에서 width와 depth의 영향
- Dataset의 크기에 비해 width or depth가 증가

  → first principal component의 similarity를 보이는 특징적인 ***block structure***가 발생
- block structure가 모델마다 고유함

  →그 외 다른 학습된 feature들은 **initialization, architecture, relative depth of the network**에 걸쳐 공유됨
- Wide and deep network는 대표적인 특성과 성능에서 유사함

  → example과 class level에서 network prediction에 다른 영향을 미침
- Open questions
  - **block structure가 학습 중에 어떻게 발생할까?**
  - **Depth and width의 의미를 활용해서 optimal task-specific model design은 어떻게 해야할까?**

감 사 합 니 다