Representations: Looking Inside Networks

COMP6258

Quick Recap

What are “learned representations”?

Learned Representations

- Informal: How the model encodes the input
- At a particular point in the network: collection of outputs for all possible inputs (often implied: belonging to your data distribution)
- In papers often people refer to feature maps of input images as the “learned representations”

Quick Recap

Why should we care about them?

Quick Recap

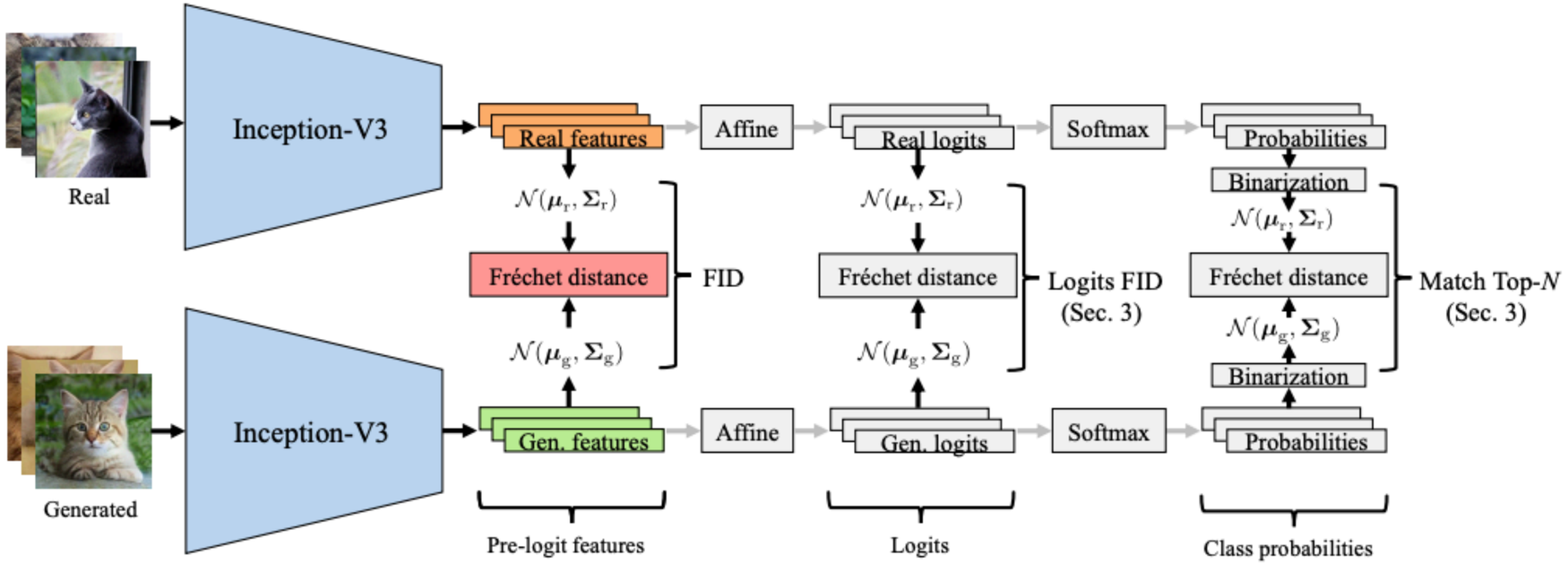
Test Accuracy Is Not Everything
(Reminder)

Quick Recap

Test Accuracy Is Not Everything
(Reminder)

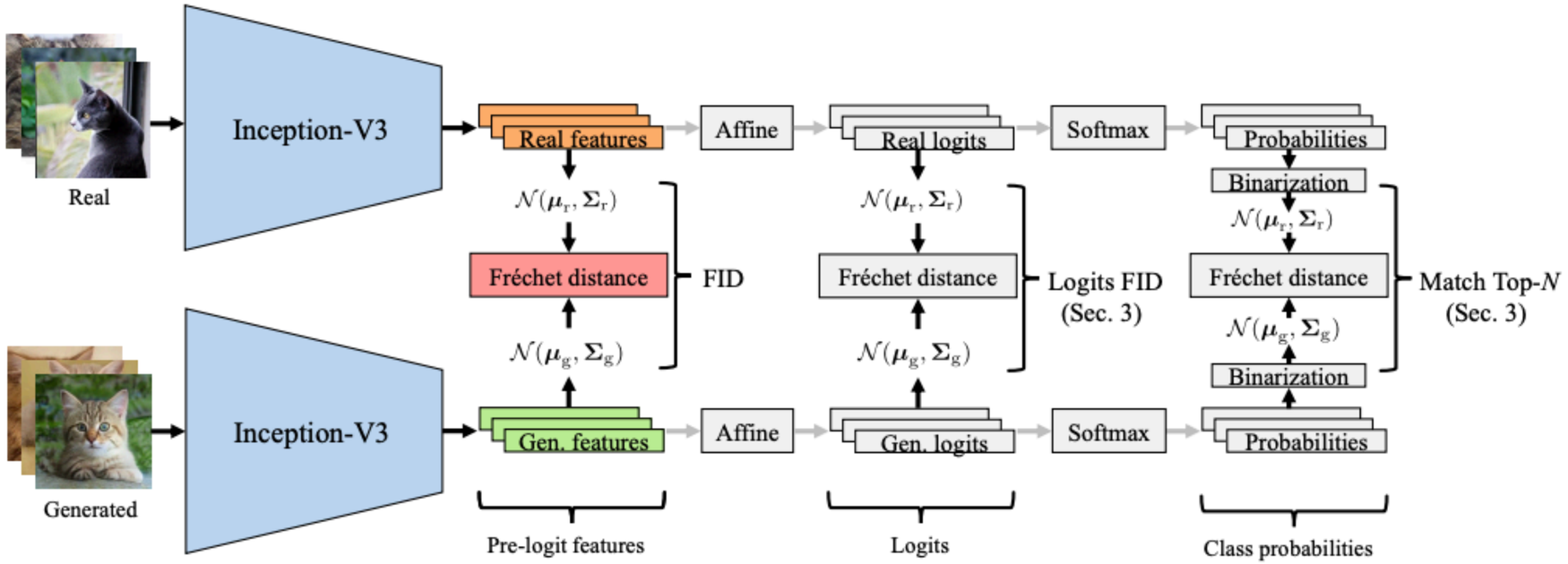
e.g. Use distance in representational space to gauge performance on unseen samples?

Quick Recap: FID



$$\mathrm{FID}\left(\boldsymbol{\mu}_{\mathrm{r}}, \boldsymbol{\Sigma}_{\mathrm{r}}, \boldsymbol{\mu}_{\mathrm{g}}, \boldsymbol{\Sigma}_{\mathrm{g}}\right) = \left\|\boldsymbol{\mu}_{\mathrm{r}} - \boldsymbol{\mu}_{\mathrm{g}}\right\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_{\mathrm{r}} + \boldsymbol{\Sigma}_{\mathrm{g}} - 2\left(\boldsymbol{\Sigma}_{\mathrm{r}}\boldsymbol{\Sigma}_{\mathrm{g}}\right)^{\frac{1}{2}}\right),$$

Quick Recap: FID



$$\mathrm{FID}\left(\boldsymbol{\mu}_{\mathrm{r}}, \boldsymbol{\Sigma}_{\mathrm{r}}, \boldsymbol{\mu}_{\mathrm{g}}, \boldsymbol{\Sigma}_{\mathrm{g}}\right) = \left\|\boldsymbol{\mu}_{\mathrm{r}} - \boldsymbol{\mu}_{\mathrm{g}}\right\|_2^2 + \mathrm{Tr}\left(\boldsymbol{\Sigma}_{\mathrm{r}} + \boldsymbol{\Sigma}_{\mathrm{g}} - 2\left(\boldsymbol{\Sigma}_{\mathrm{r}}\boldsymbol{\Sigma}_{\mathrm{g}}\right)^{\frac{1}{2}}\right),$$

We need to understand the representational spaces

Let’s look at some feature maps

Let’s look at some feature maps

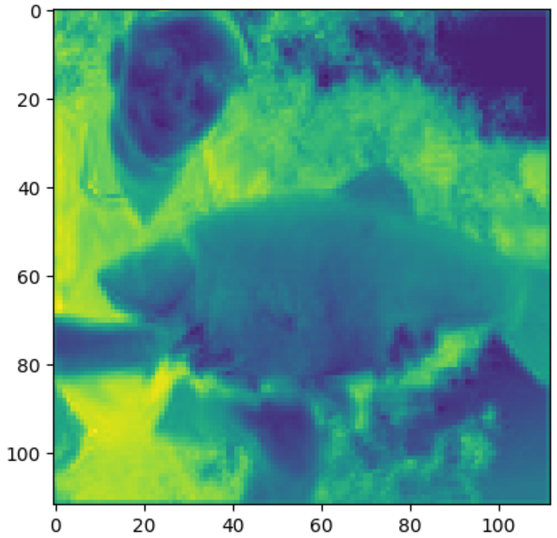
(First feature map for a few different samples)

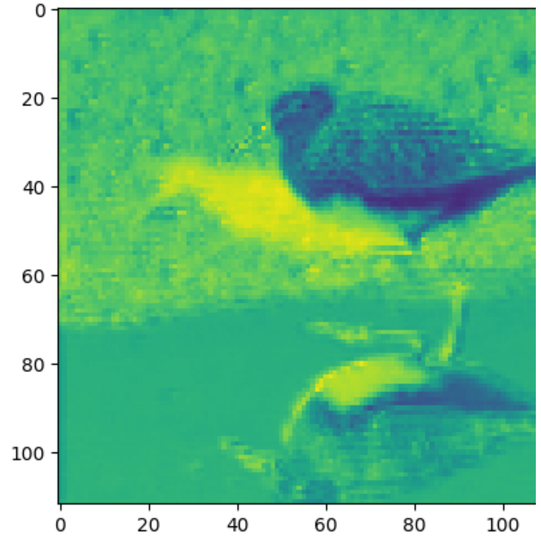
ResNet-18 trained on ImageNet

Feature maps after the first convolutional layer

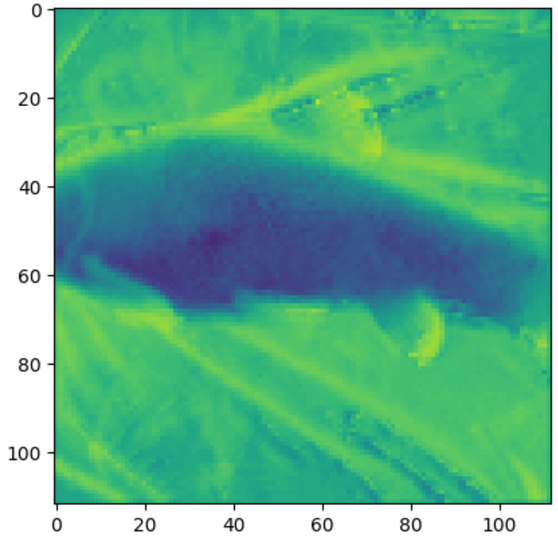
Reference

feature map

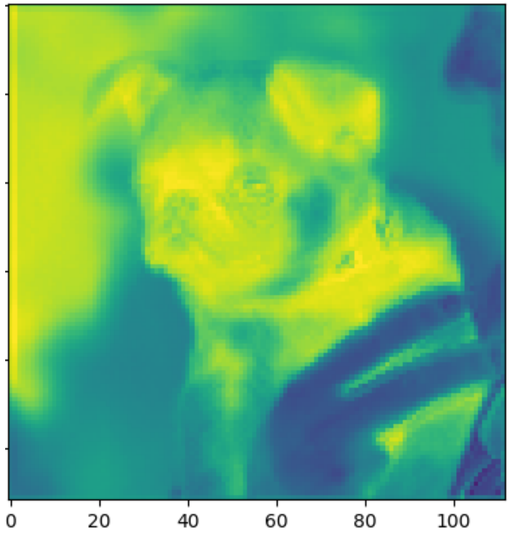




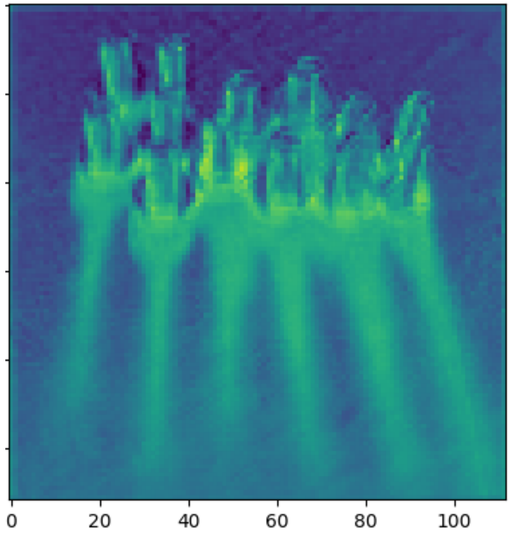
A



B



C



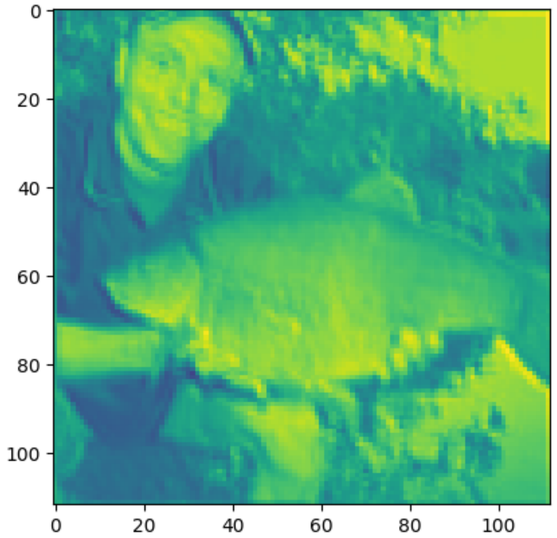
D

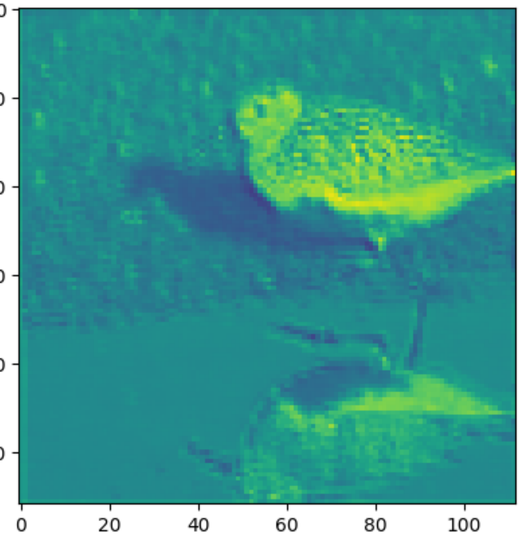
Let’s look at some feature maps

(Second feature map for a few different samples)

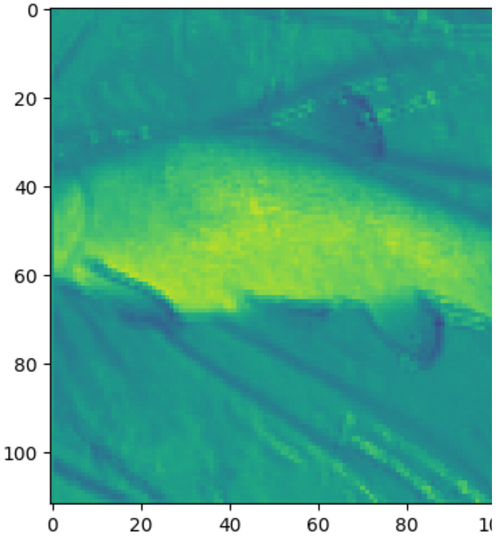
Reference

feature map

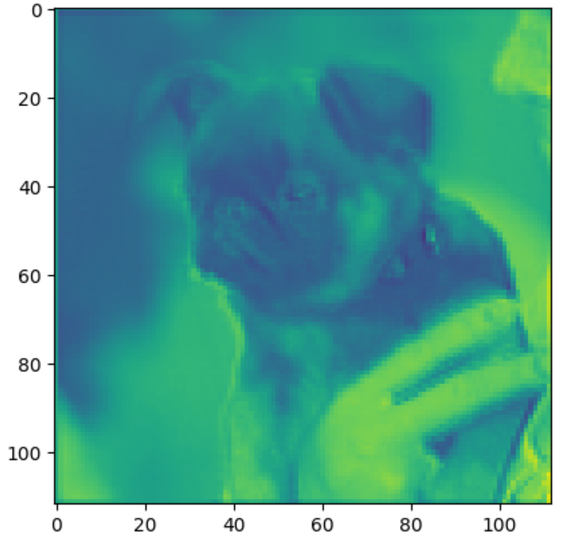




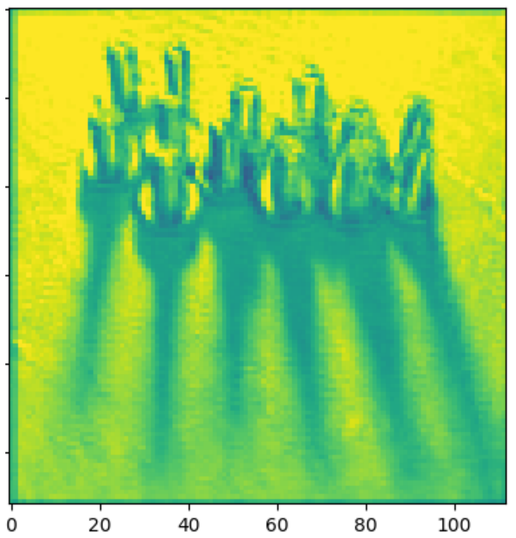
A



B



C

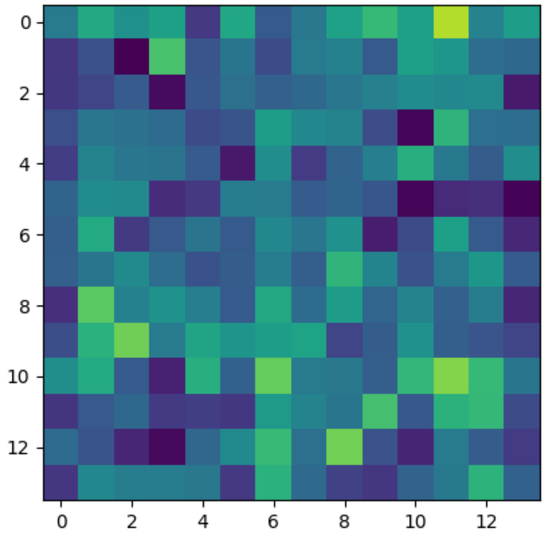


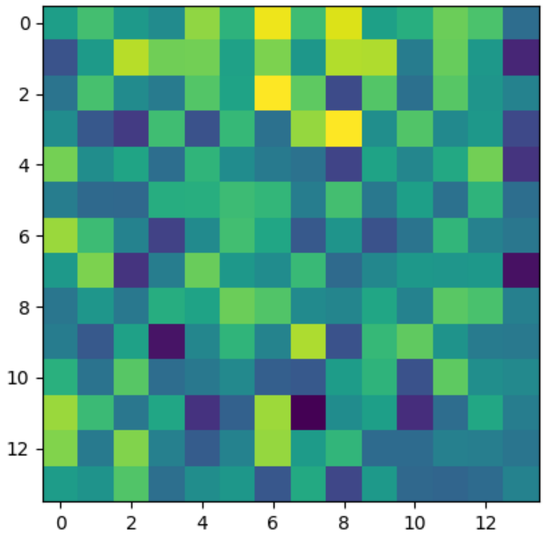
D

Now let’s look at a later point in the network

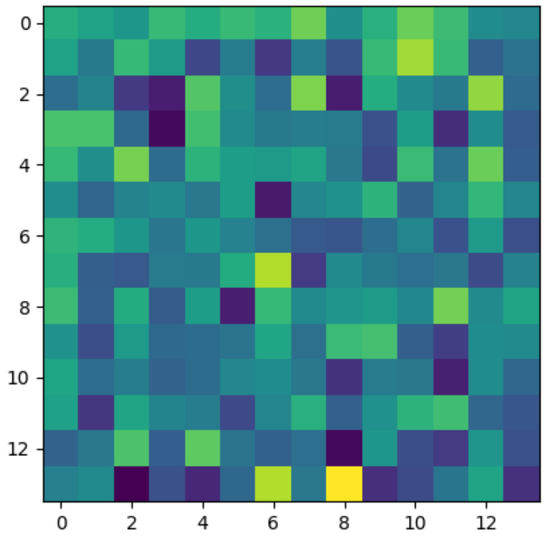
Reference

feature map

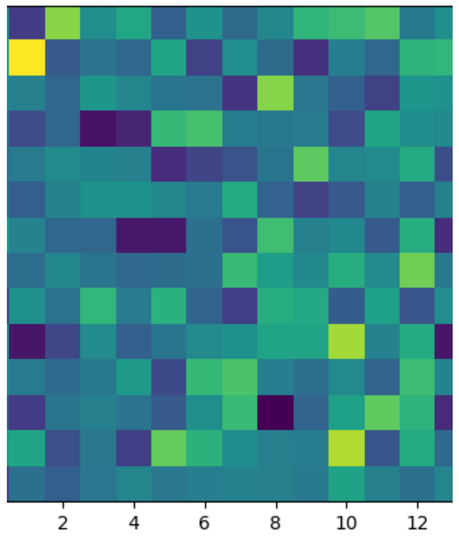




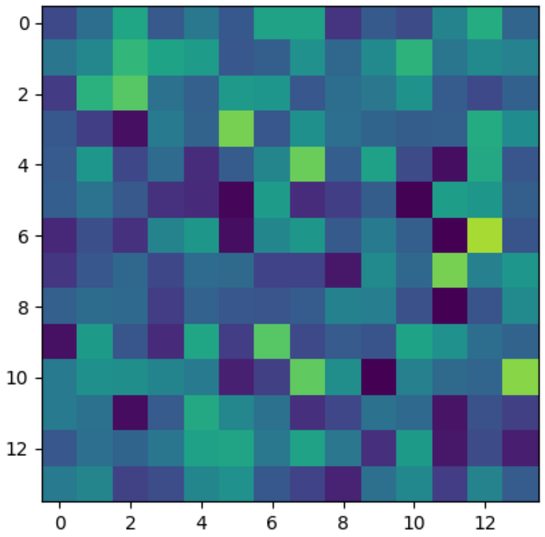
A



B



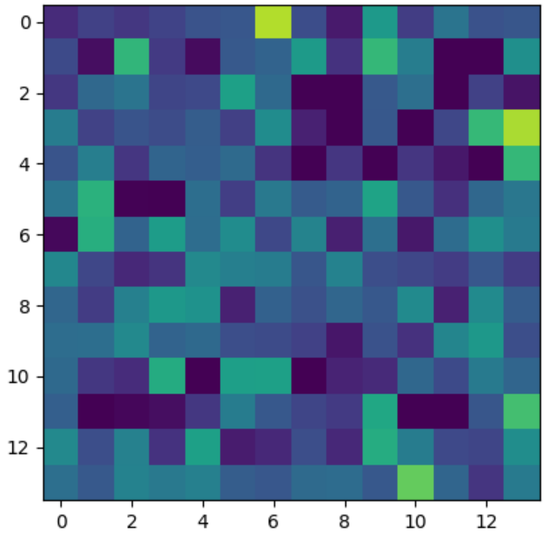
C

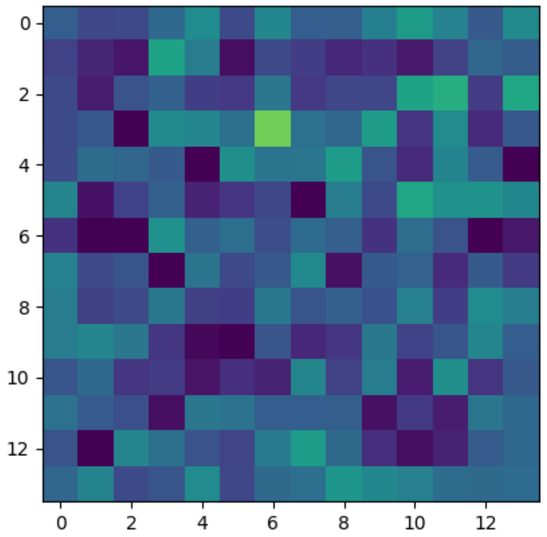


D

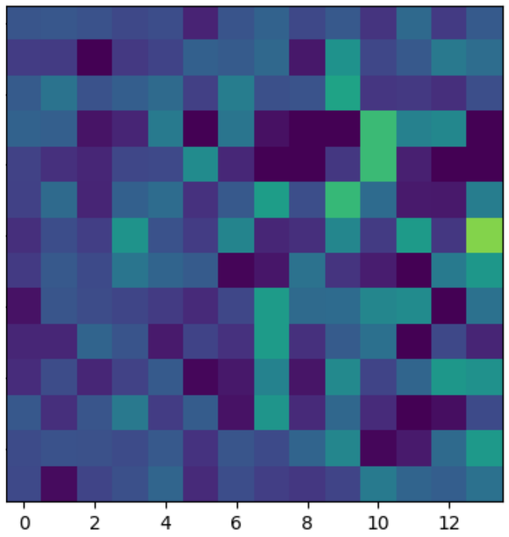
Reference

feature map

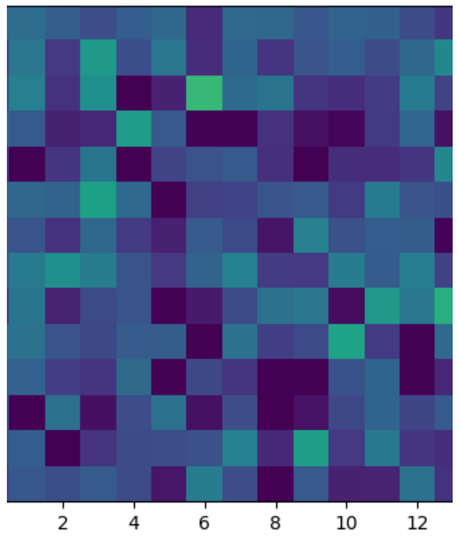




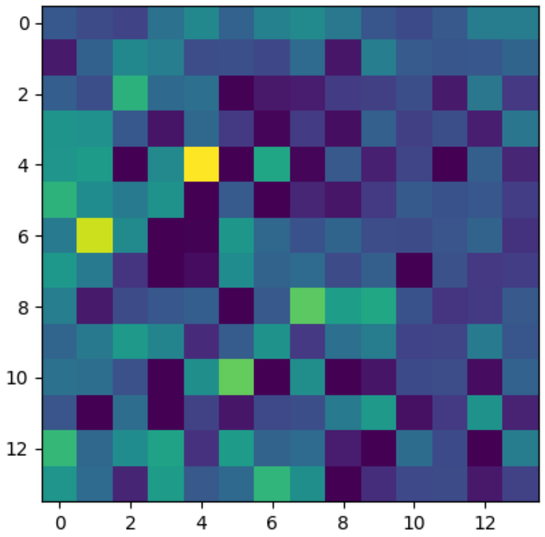
A



B



C



D

Why is it hard to identify similarity?

(MANY reasons, but let’s give some examples)

How can we find out if similar input patterns are encoded by a collection of feature maps?

How can we find out if similar input patterns are encoded by a collection of feature maps?

functional

How can we find out if similar input patterns are encoded by a collection of feature maps?

functional

structural

How can we find out if similar input patterns are encoded by a collection of feature maps?

functional

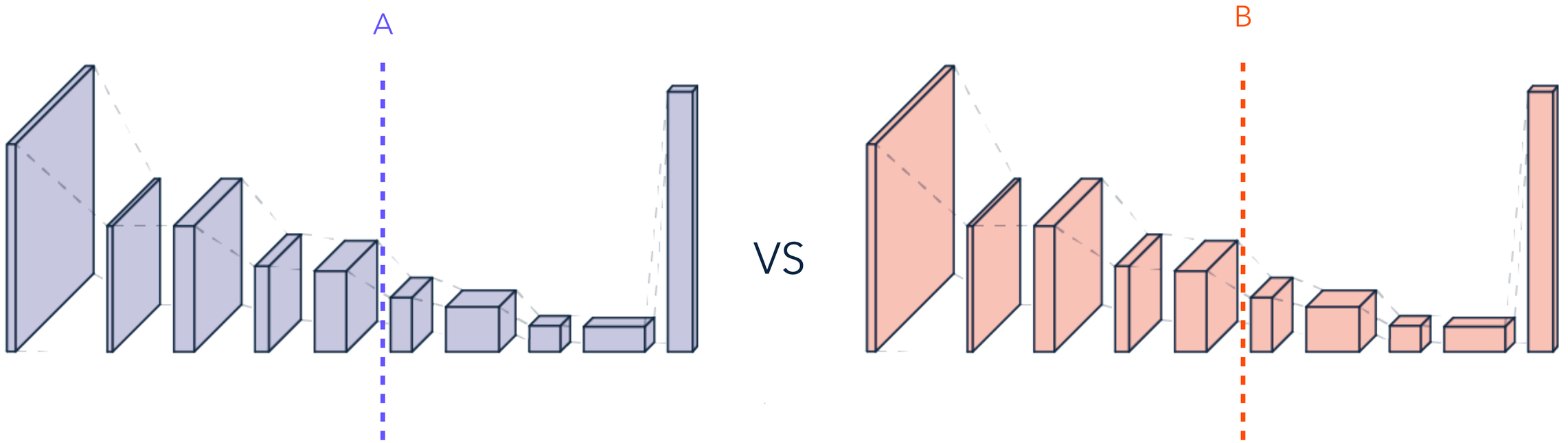
structural

Model Stitching

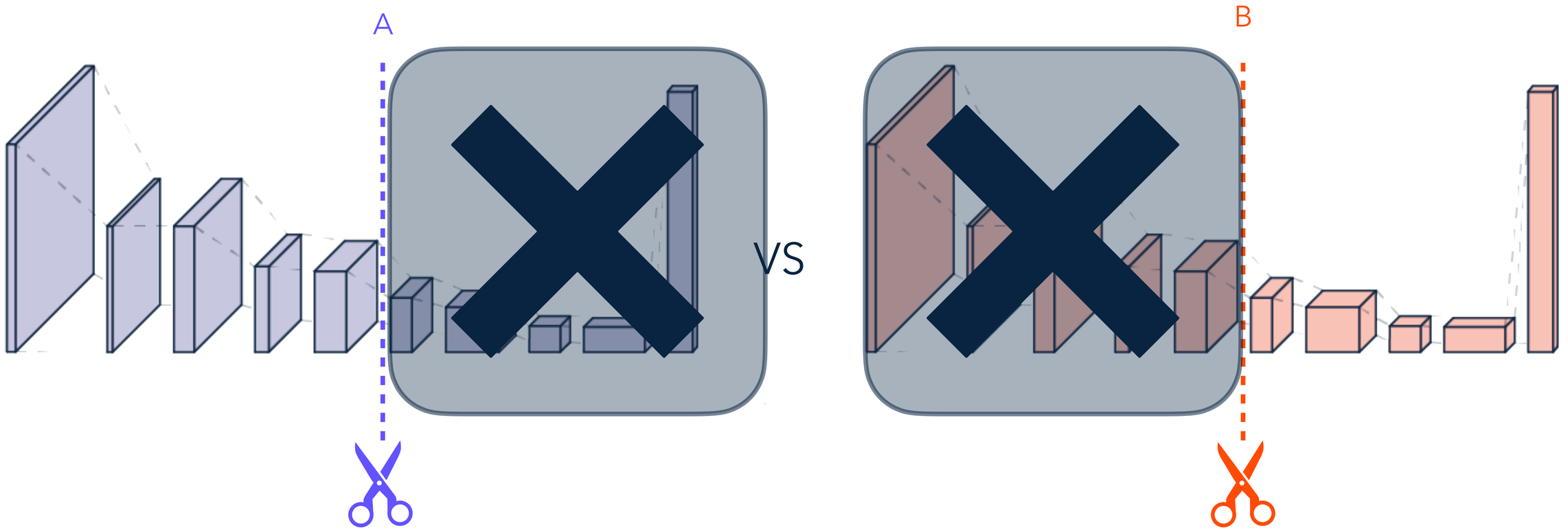
Model Stitching

- Intuition: Using certain feature maps, how well can a model **perform**?

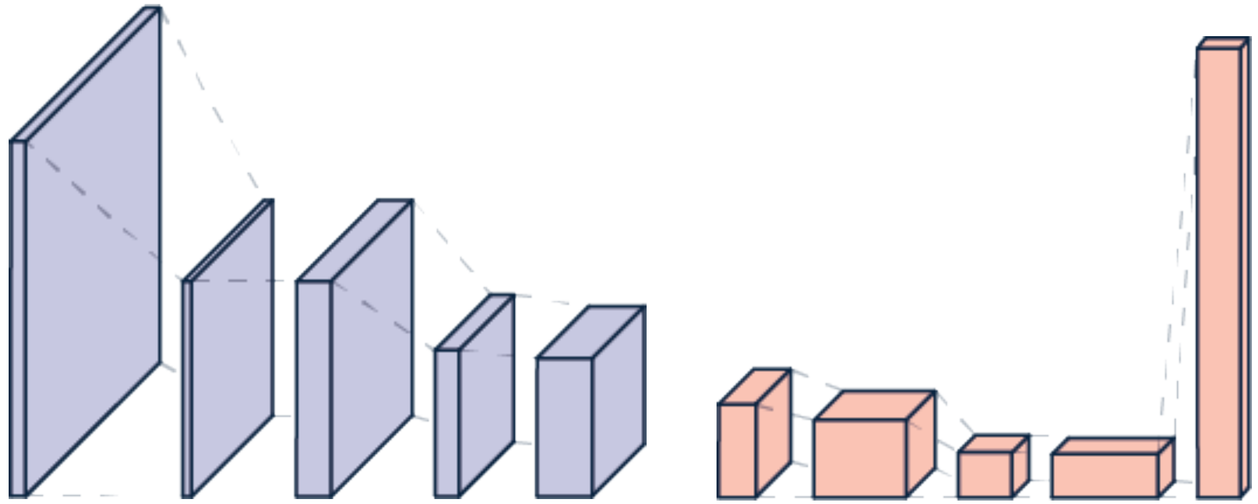
Model Stitching



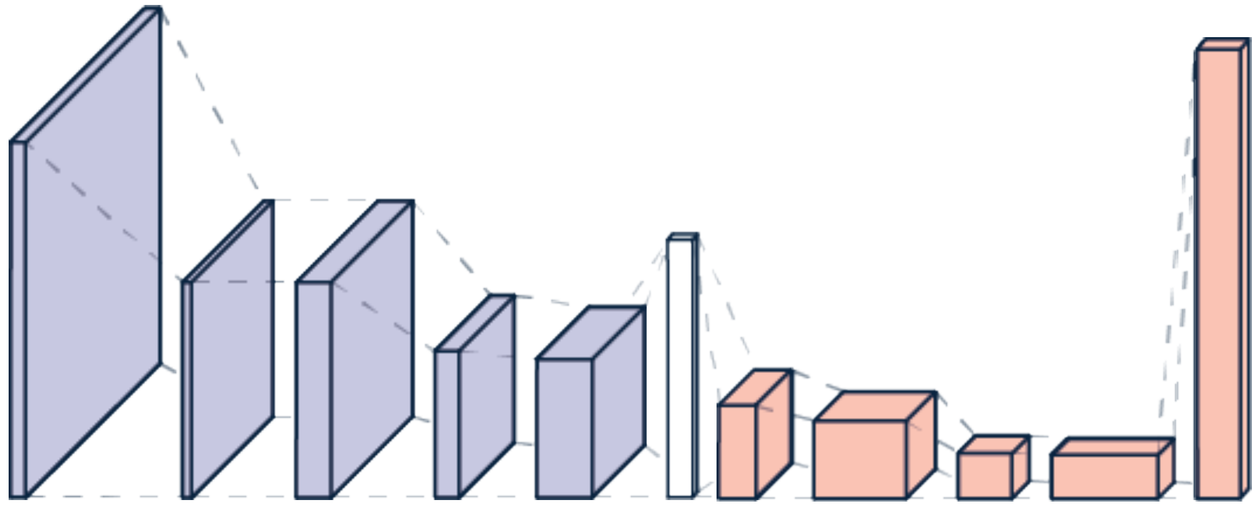
Model Stitching



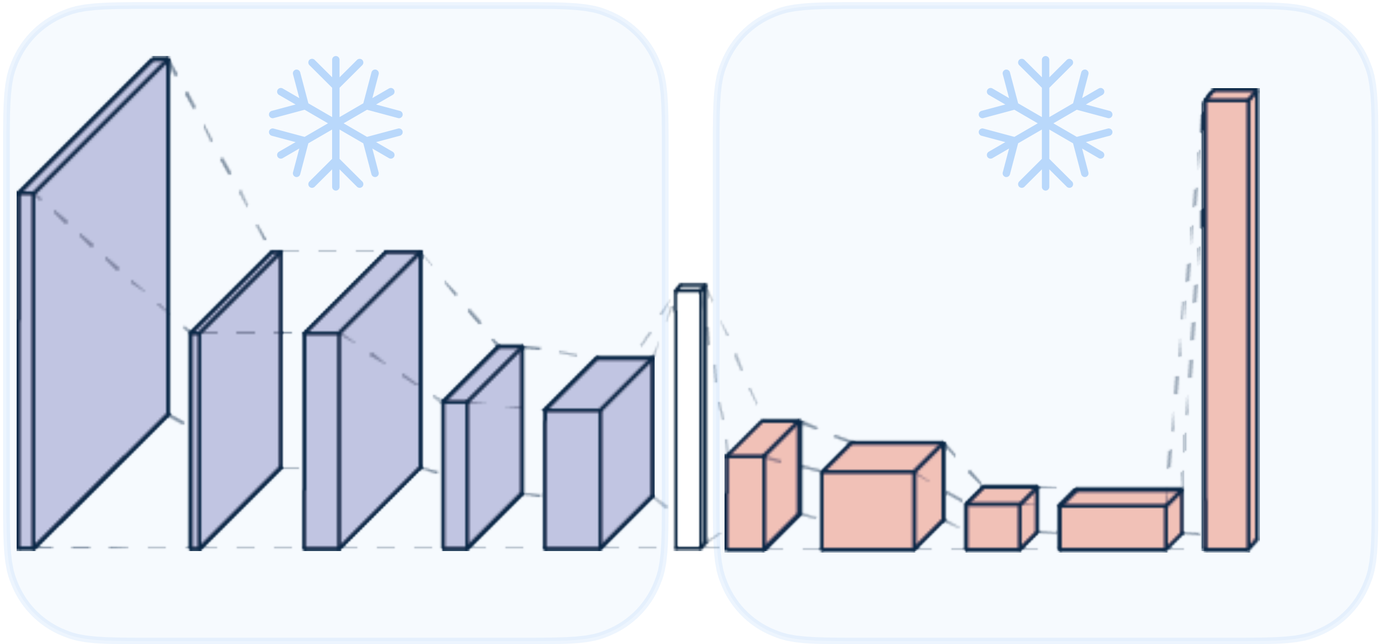
Model Stitching



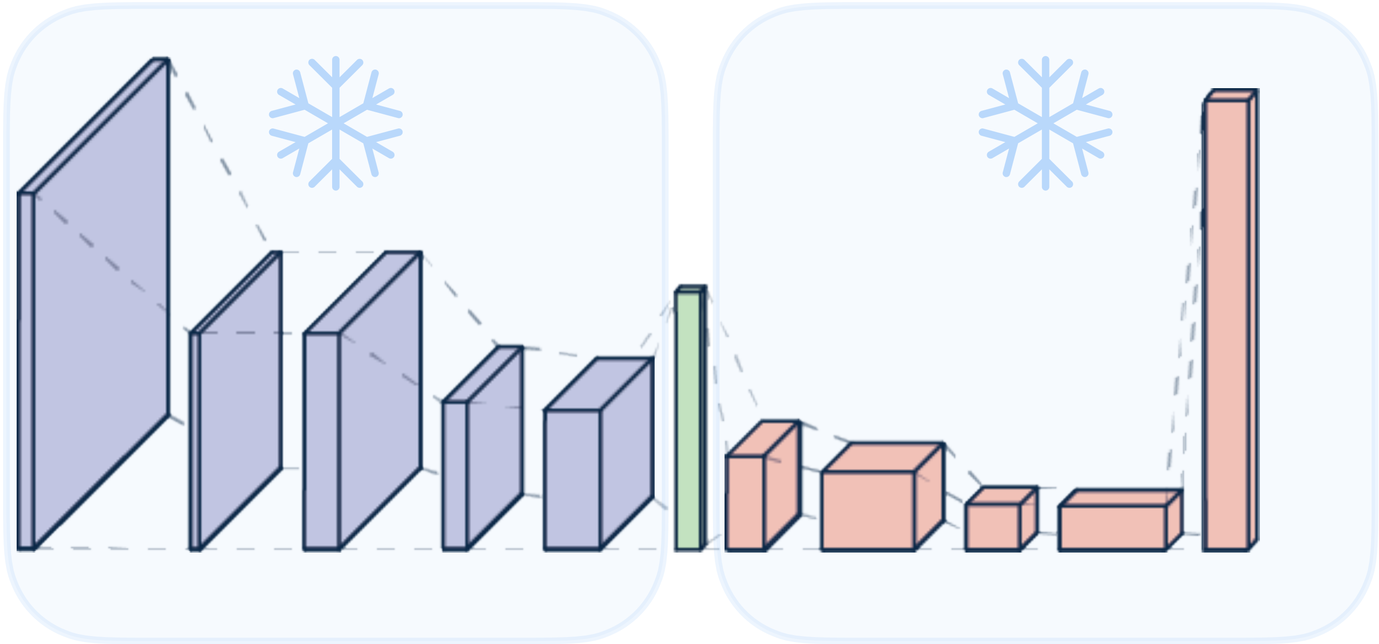
Model Stitching



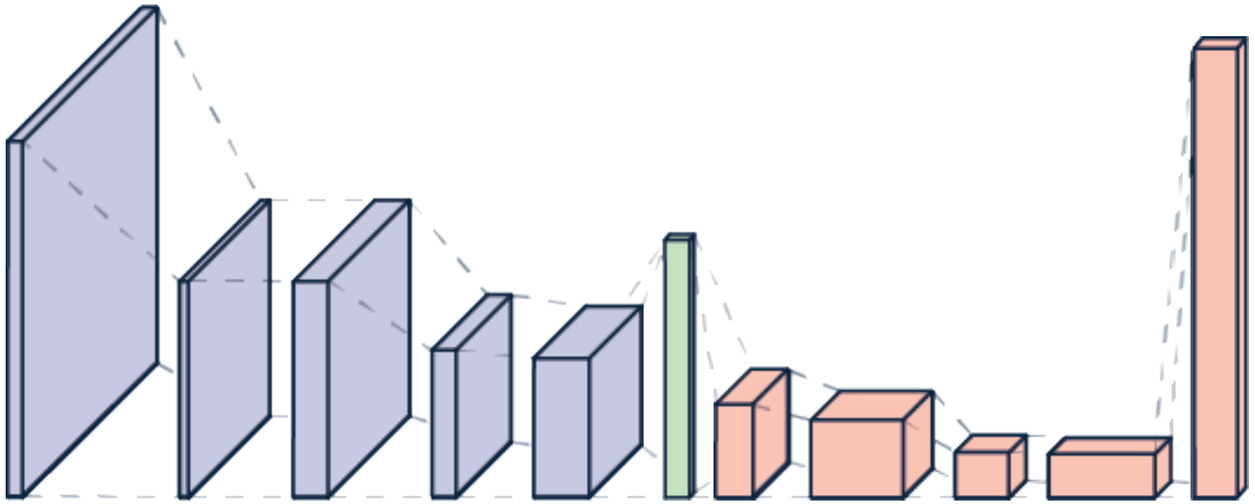
Model Stitching



Model Stitching



Model Stitching



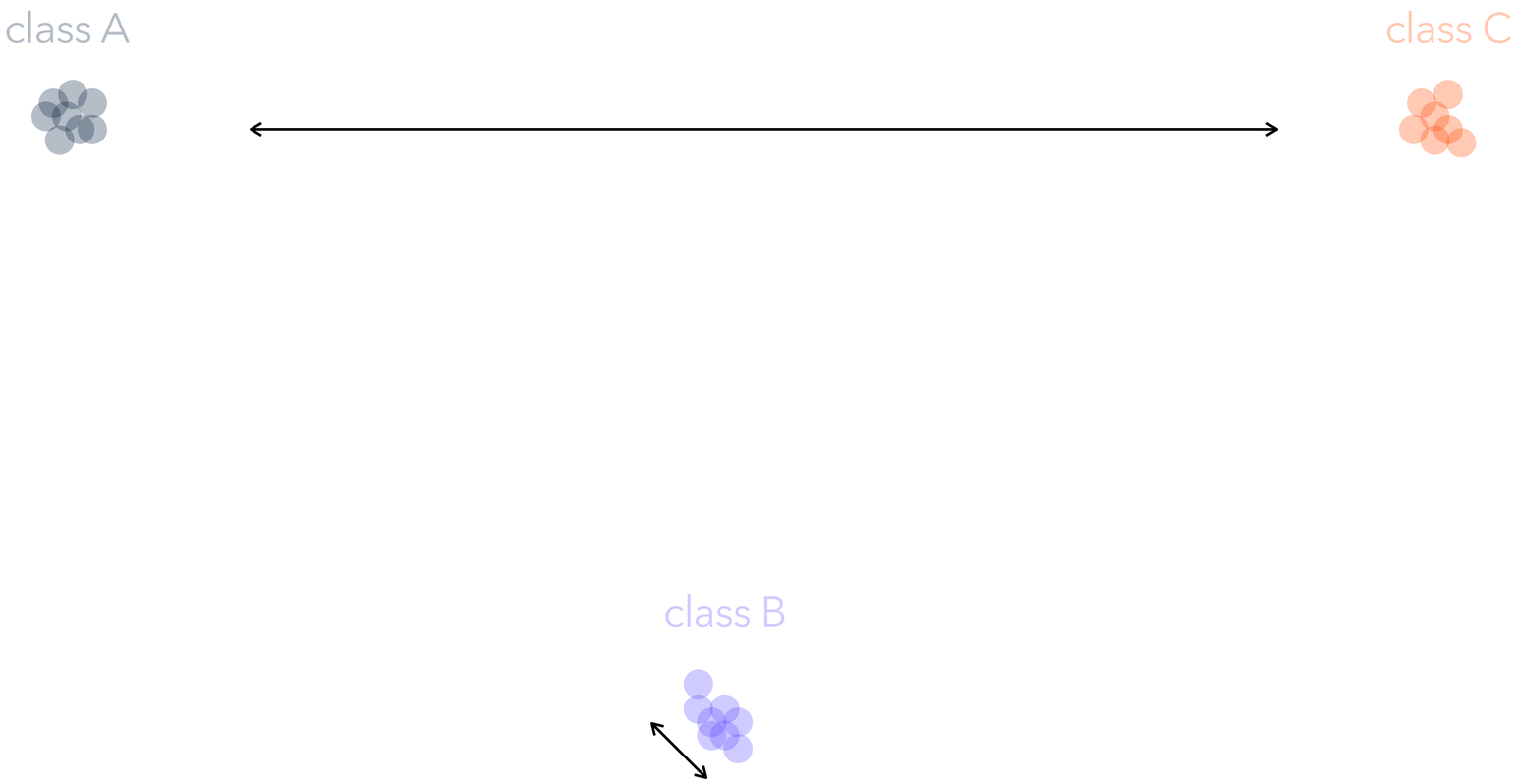
Can this model perform well?
If yes, representations A and B are “compatible”

Reminder: How can we find out if similar input patterns are encoded by a collection of feature maps?

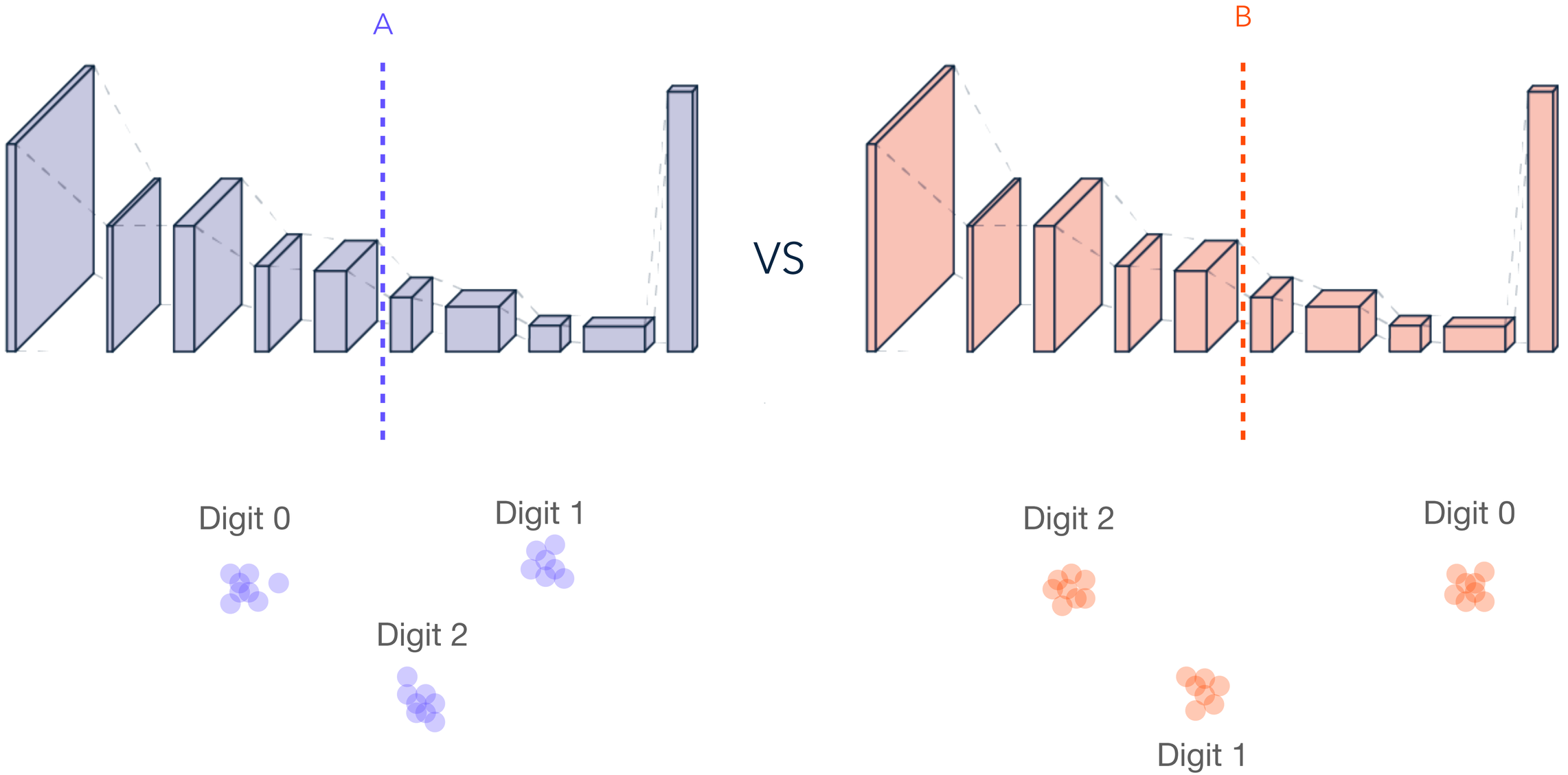
Reminder: How can we find out if similar input patterns are encoded by a collection of feature maps?

What does compatibility of embeddings tell us about input patterns represented?

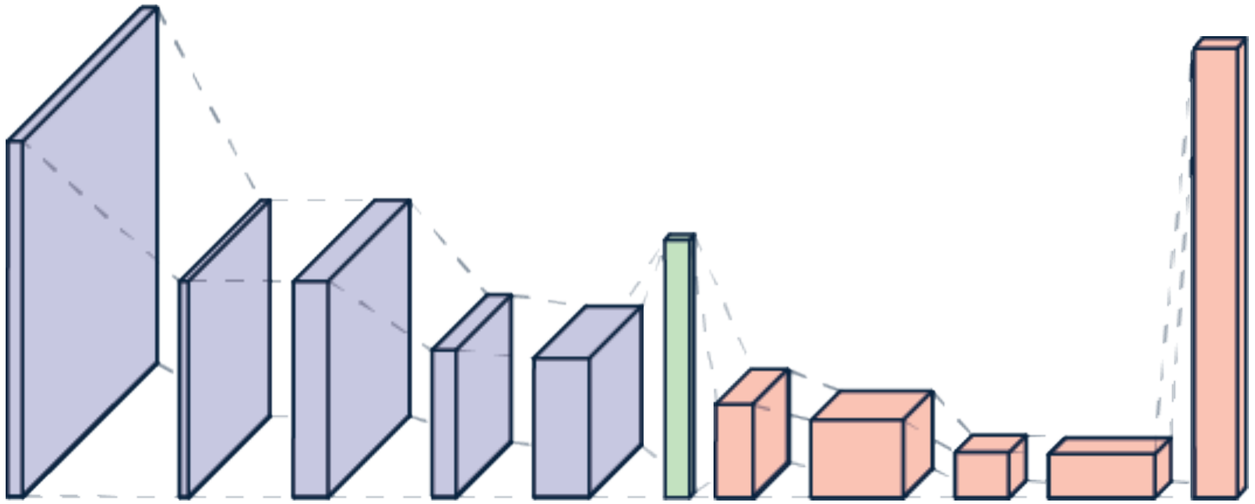
Reminder

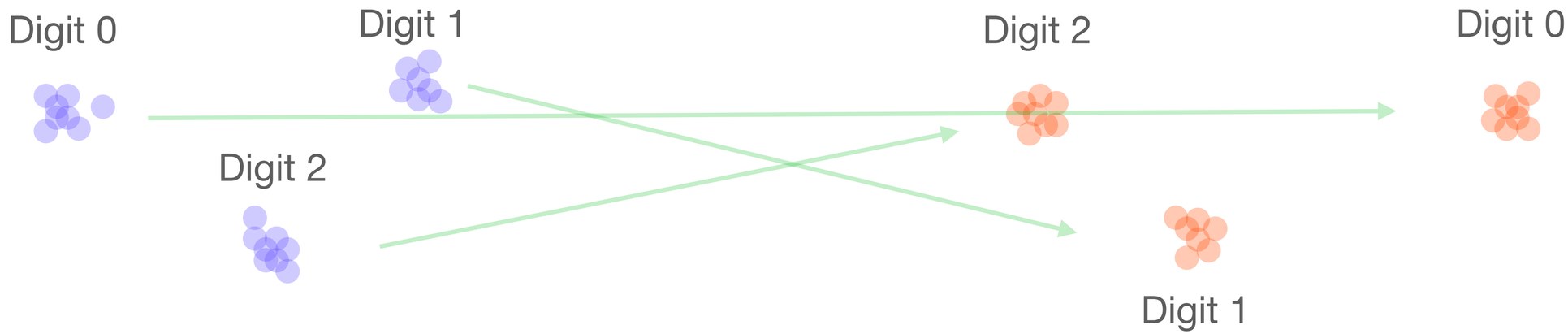


Model Stitching

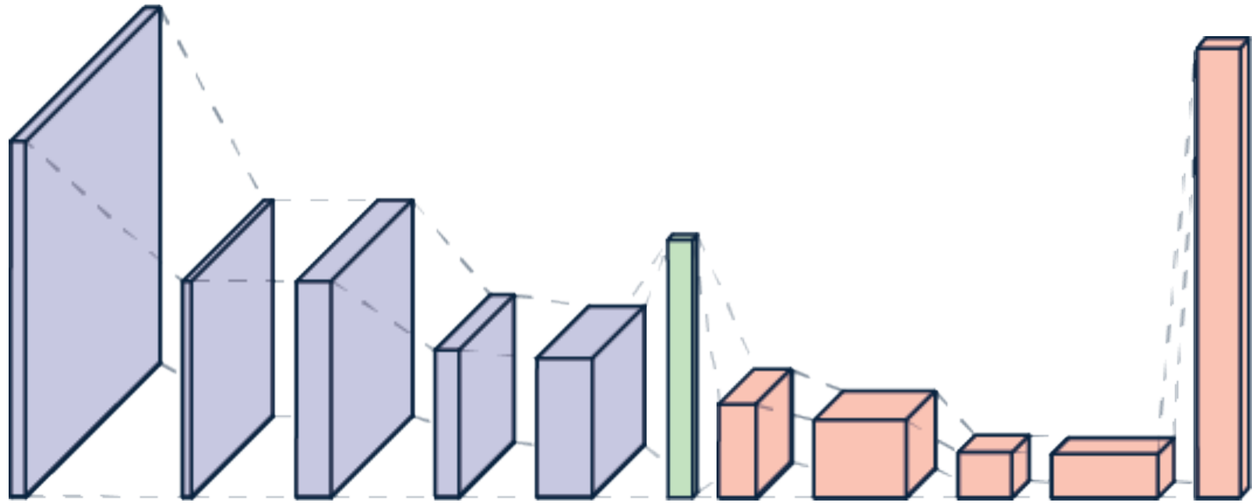


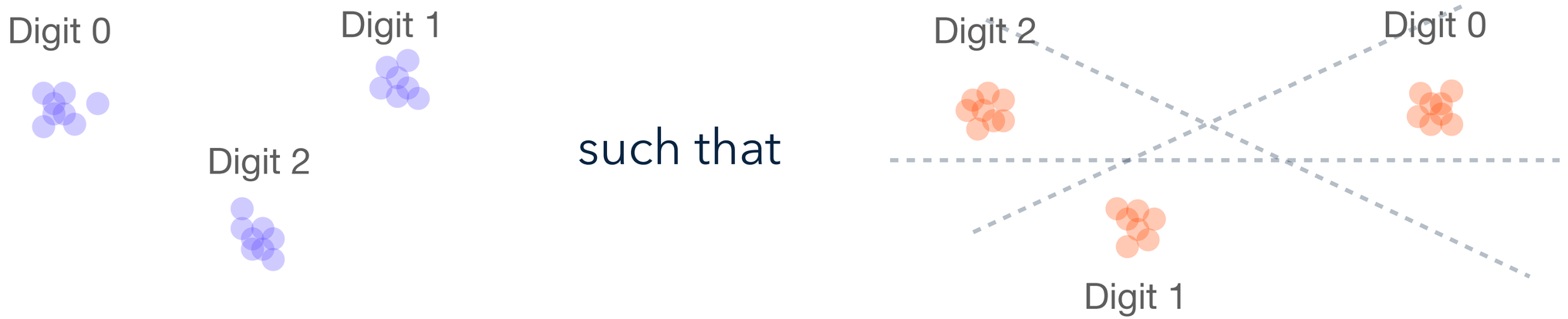
Model Stitching



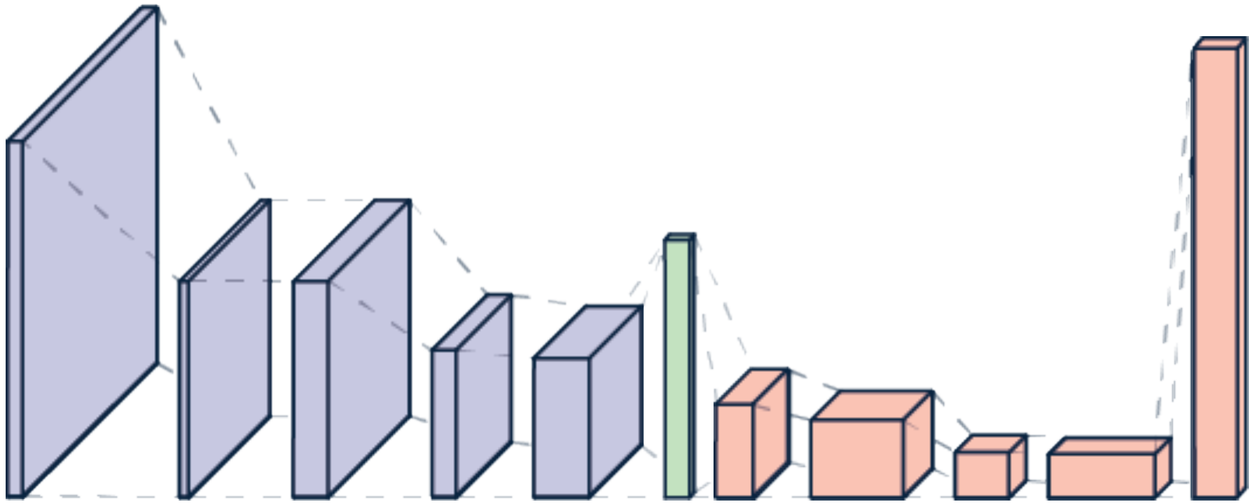


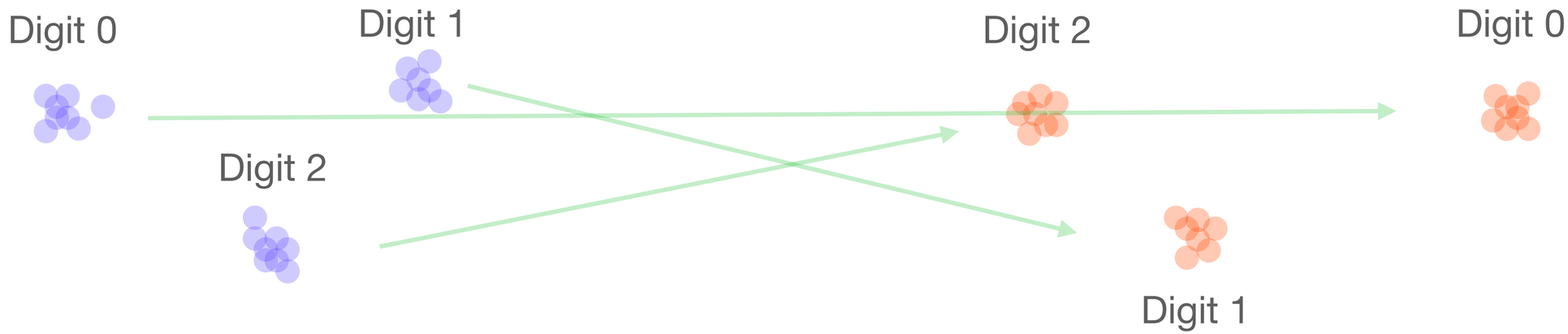
Model Stitching



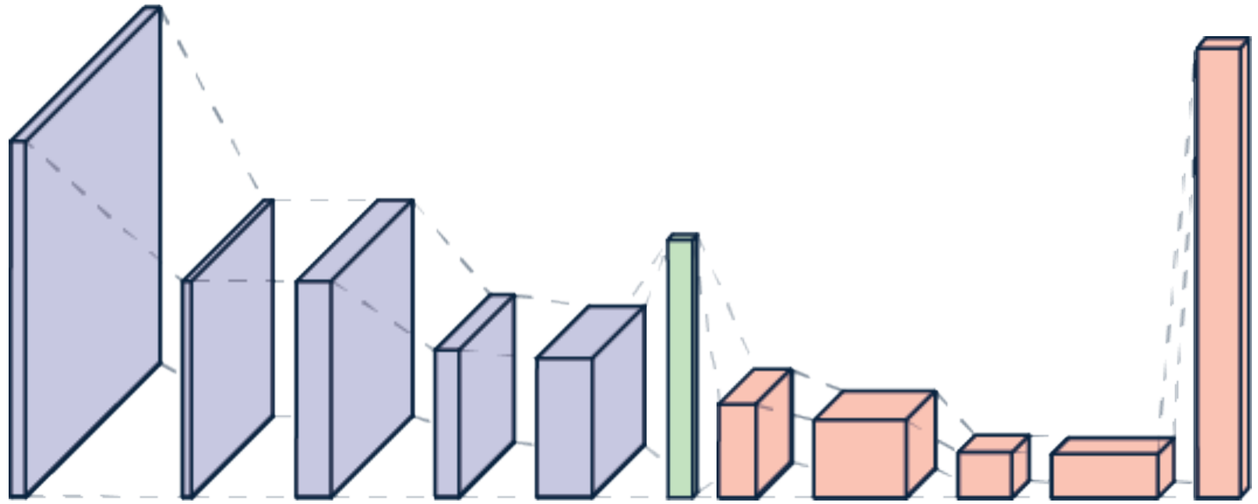


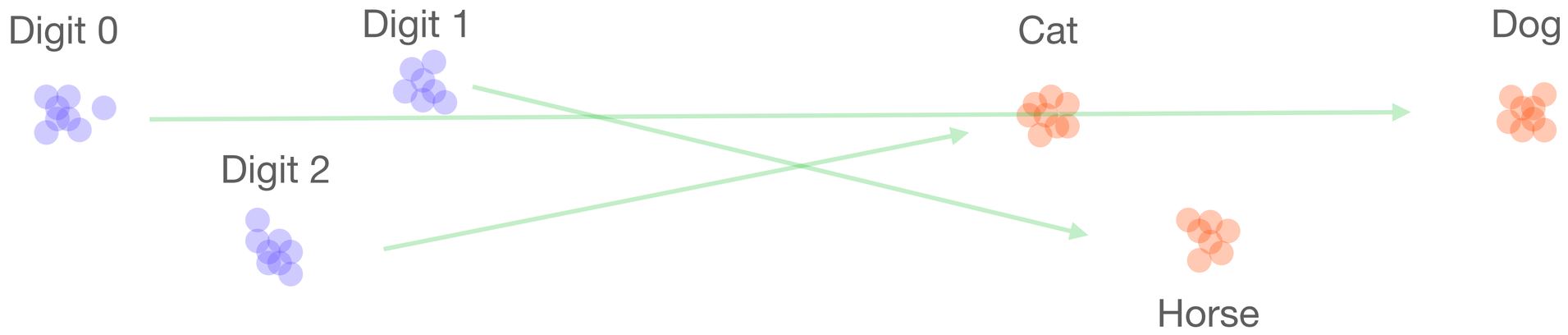
Model Stitching



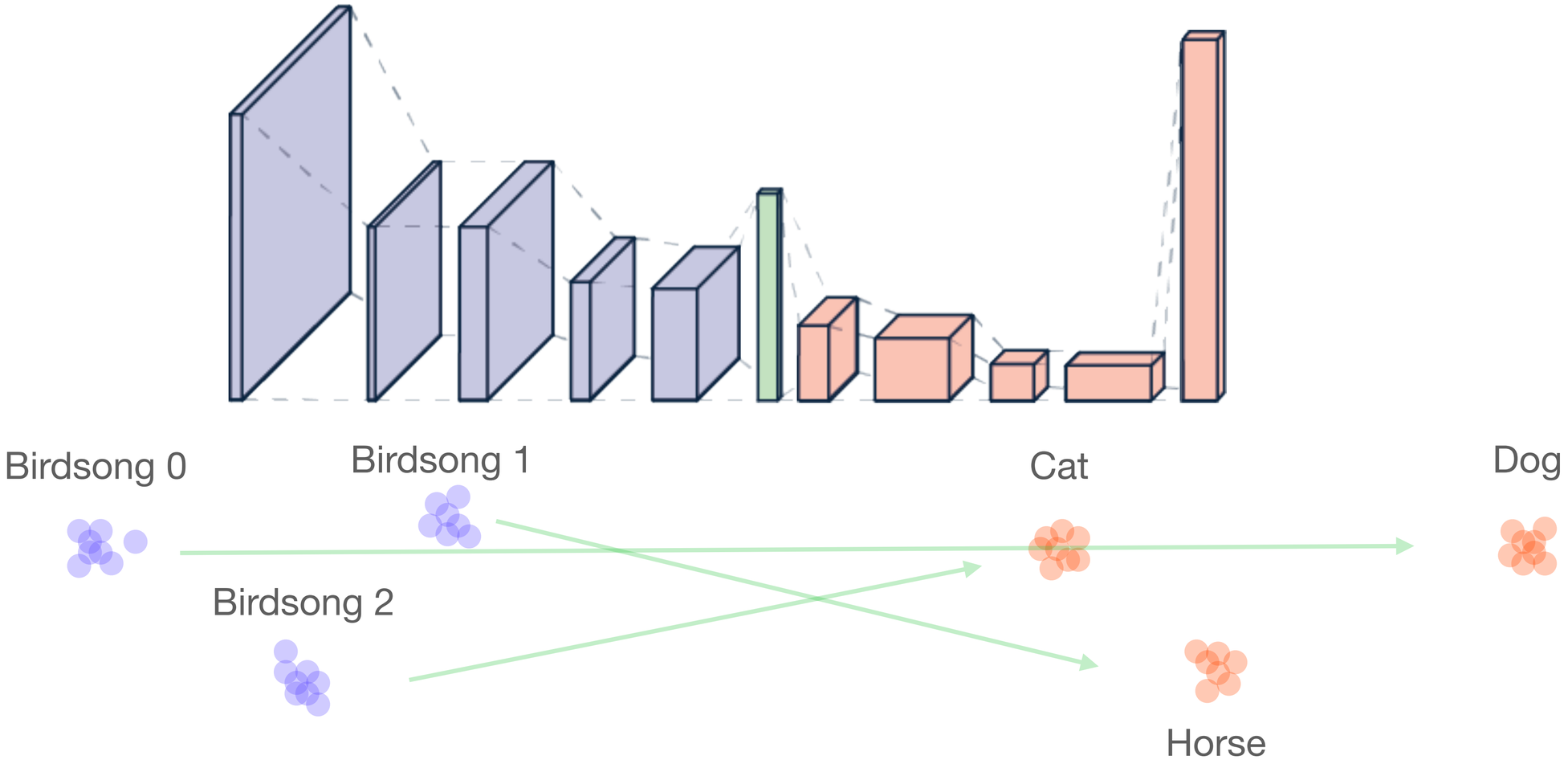


Model Stitching





Model Stitching



Platonic Representations?

How can we find out if similar input patterns are encoded by a collection of feature maps?

functional

structural

Centered Kernel Alignment (CKA)

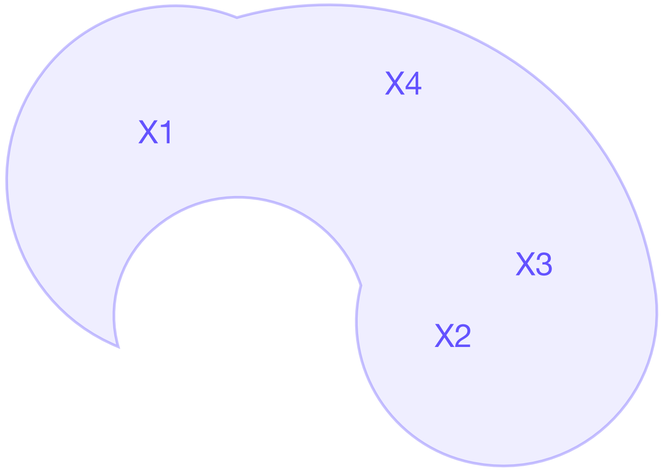
- Most common method

Centered Kernel Alignment (CKA)

- Most common method
- Statistical perspective

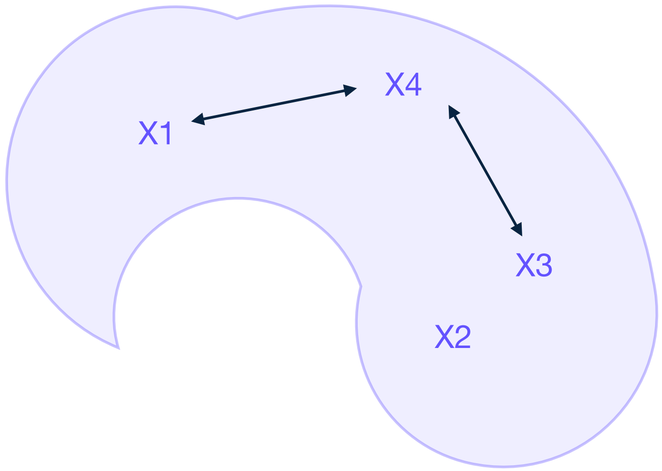
Centered Kernel Alignment (CKA)

- Most common method
- Statistical perspective
- Intuition:



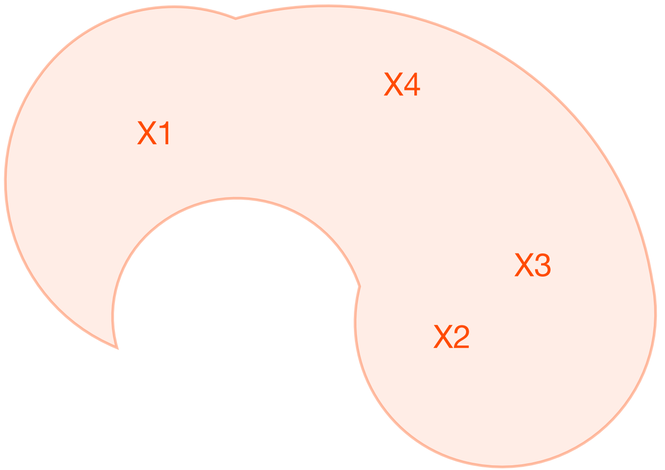
Centered Kernel Alignment (CKA)

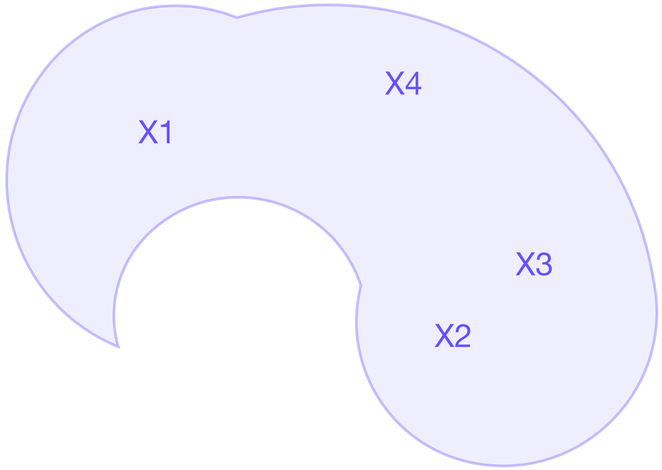
- Most common method
- Statistical perspective
- Intuition:



Centered Kernel Alignment (CKA)

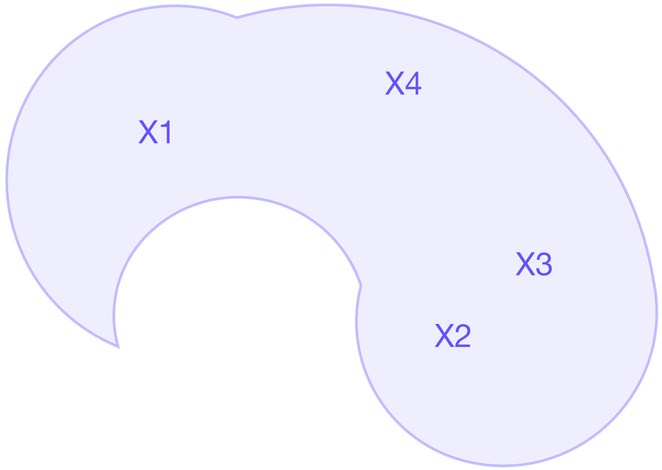
- Most common method
- Statistical perspective
- Intuition:

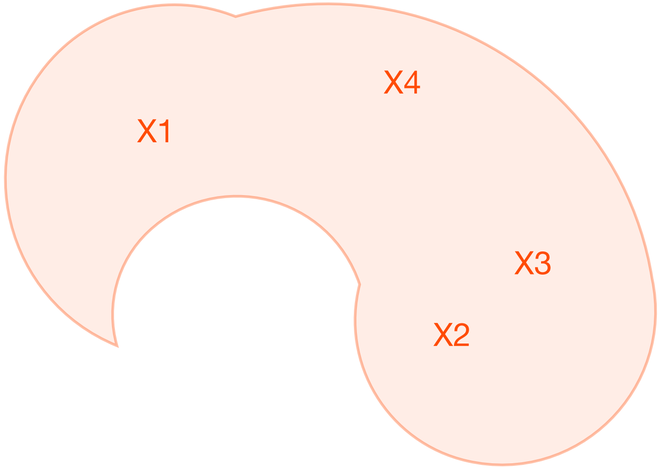




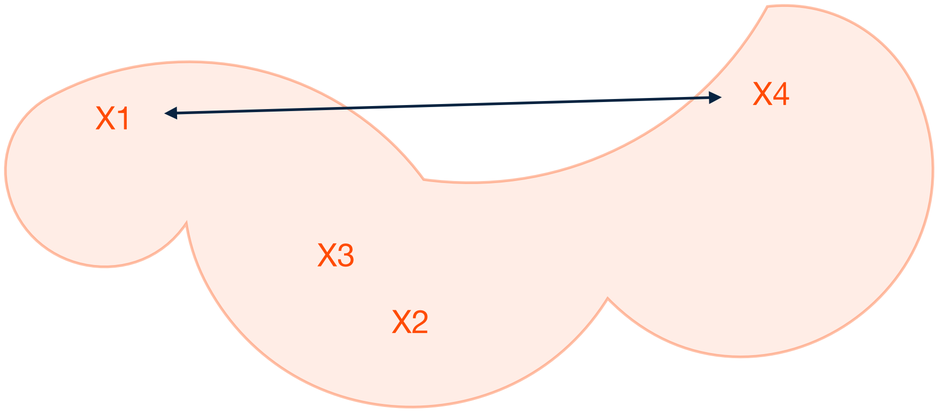
Centered Kernel Alignment (CKA)

- Most common method
- Statistical perspective
- Intuition:



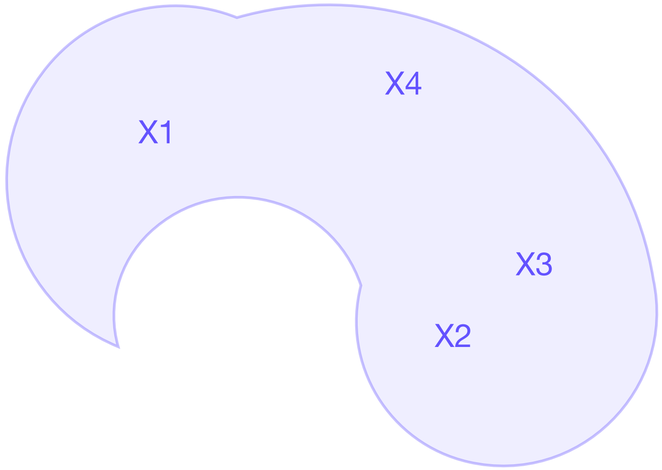


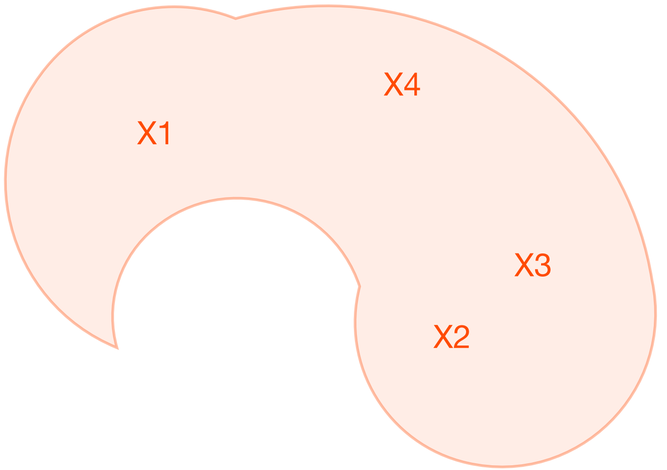
OR



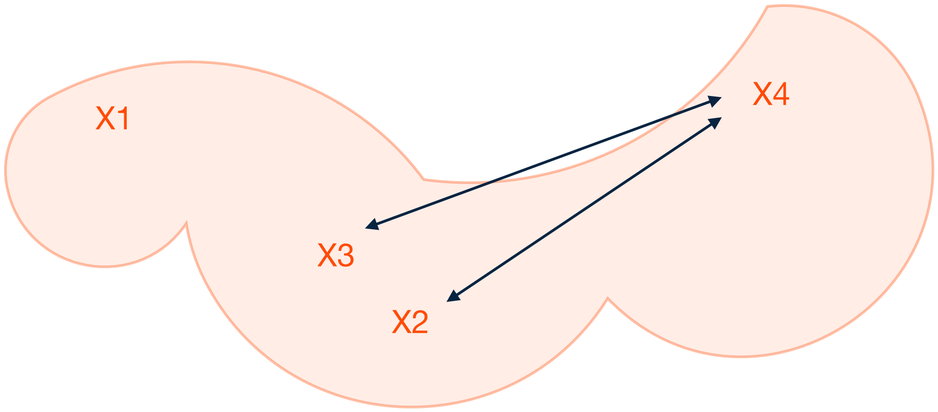
Centered Kernel Alignment (CKA)

- Most common method
- Statistical perspective
- Intuition:





OR



Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

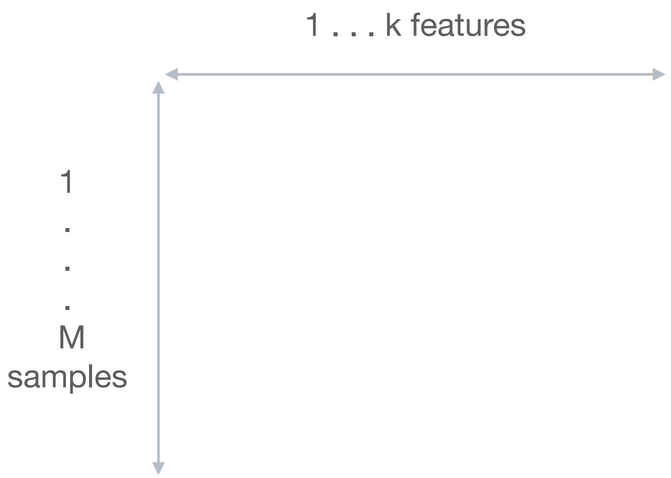
$$X = AA^{\mathsf{T}} \qquad Y = BB^{\mathsf{T}}$$

Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

$$X = AA^{\mathsf{T}}$$

$$Y = BB^{\mathsf{T}}$$



Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

$$X = AA^\top \qquad Y = BB^\top$$

M x M tensors

M - number of samples

Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

$$X = AA^{\mathsf{T}} \qquad Y = BB^{\mathsf{T}}$$

M x M tensors

M - number of samples

Similarity between each pair of samples according to all the feature maps

Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

$$X = AA^{\mathsf{T}} \qquad Y = BB^{\mathsf{T}}$$

- Hilbert-Schmidt Independence Criterion (HSIC):

$$\text{HSIC}(X, Y) = \frac{1}{(M-1)^2} \text{tr}(X^*, Y^*)$$

Centered Kernel Alignment (CKA)

- A and B are mapped to the X and Y space

$$X = AA^\top \qquad\qquad Y = BB^\top$$

- Hilbert-Schmidt Independence Criterion (HSIC):

$$\text{HSIC}(X, Y) = \frac{1}{(M-1)^2}\,\text{tr}(X^*, Y^*)$$

- Centered Kernel Alignement (CKA):

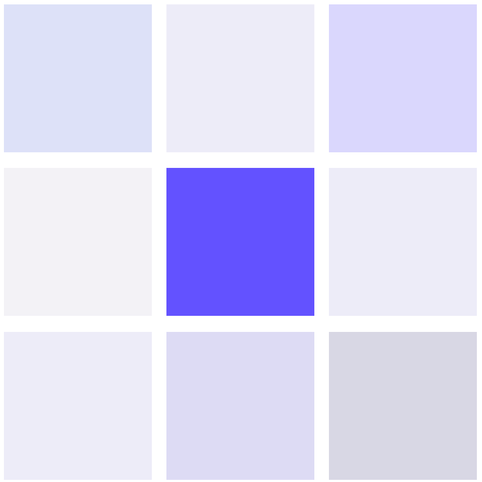
$$\text{CKA}(X, Y) = \frac{\text{HSIC}(X, Y)}{\sqrt{\text{HSIC}(X, X)\,\text{HSIC}(Y, Y)}}$$

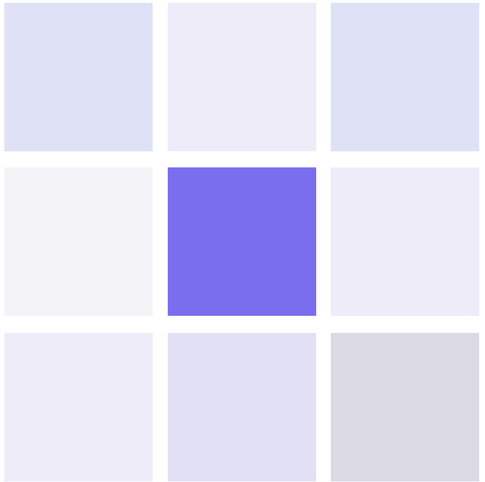
What did the functional perspective (try to) capture that CKA doesn’t?

CKA take-away

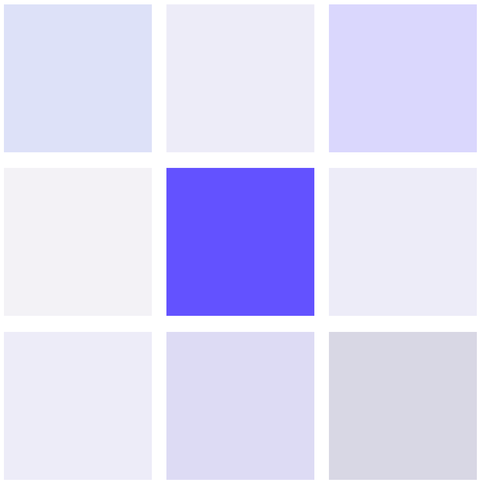
- Structural similarity is more nuanced and useful than functional similarity
- But not accounting for the functional use of the feature maps is a major drawback of structural similarity
- Compression is problematic for both perspectives

Do these feature maps capture the same patterns in the input?

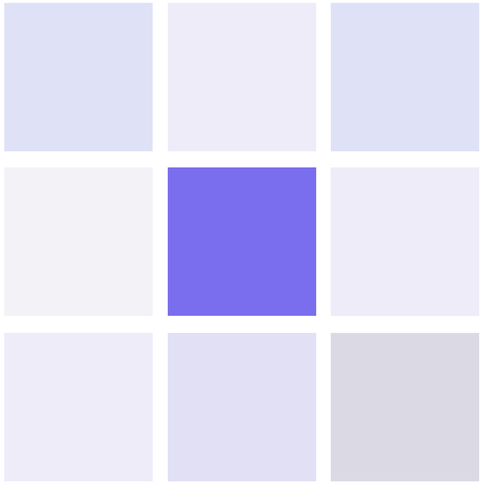


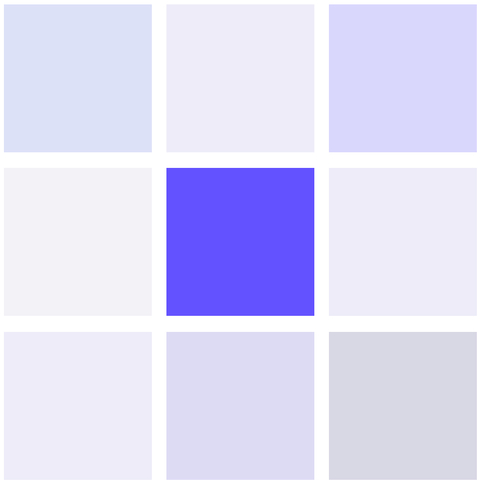


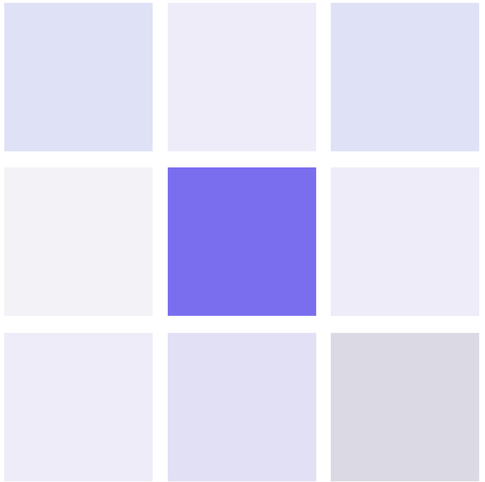
8



3







How can we find out what input patterns a feature map encodes?

Saliency Map Comparison

GradCAM

- Intent: What input regions are “important” when making a prediction?

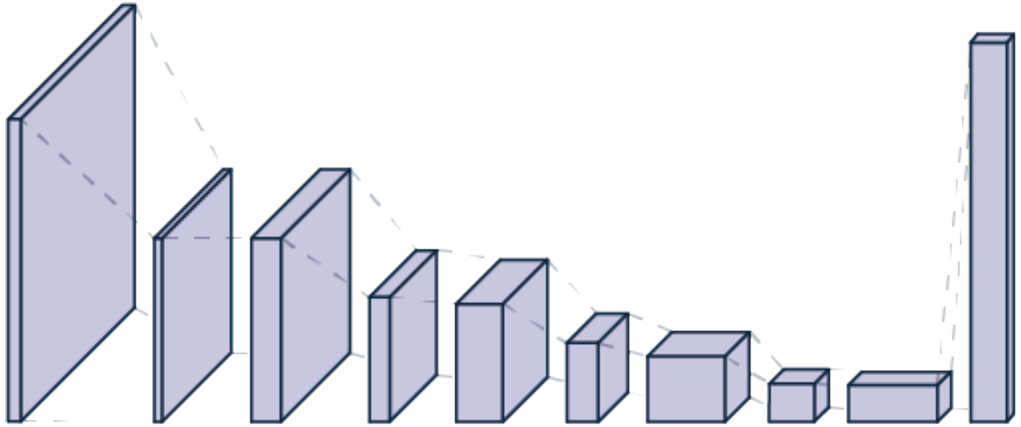
Saliency Map Comparison

GradCAM

- Intent: What input regions are “important” when making a prediction?

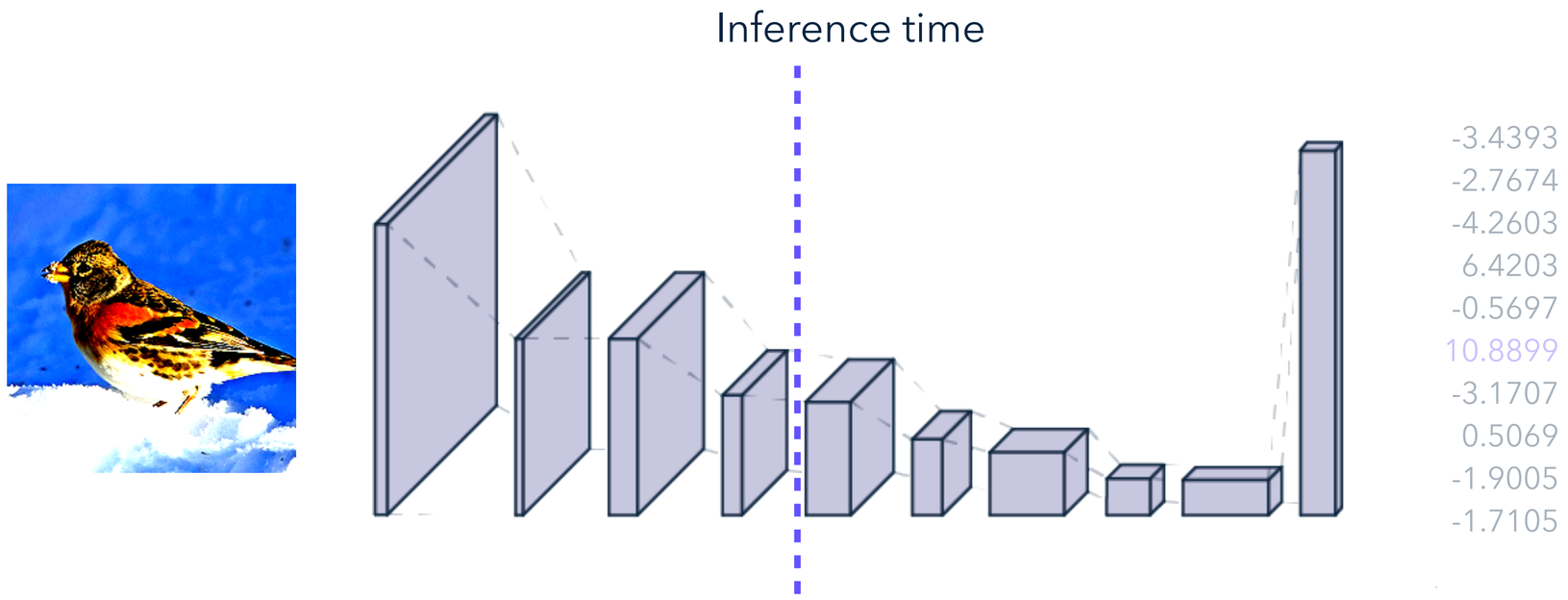
- If I make a small change to the input, how much does it affect the prediction?

Saliency Map - CIFAR10 Example

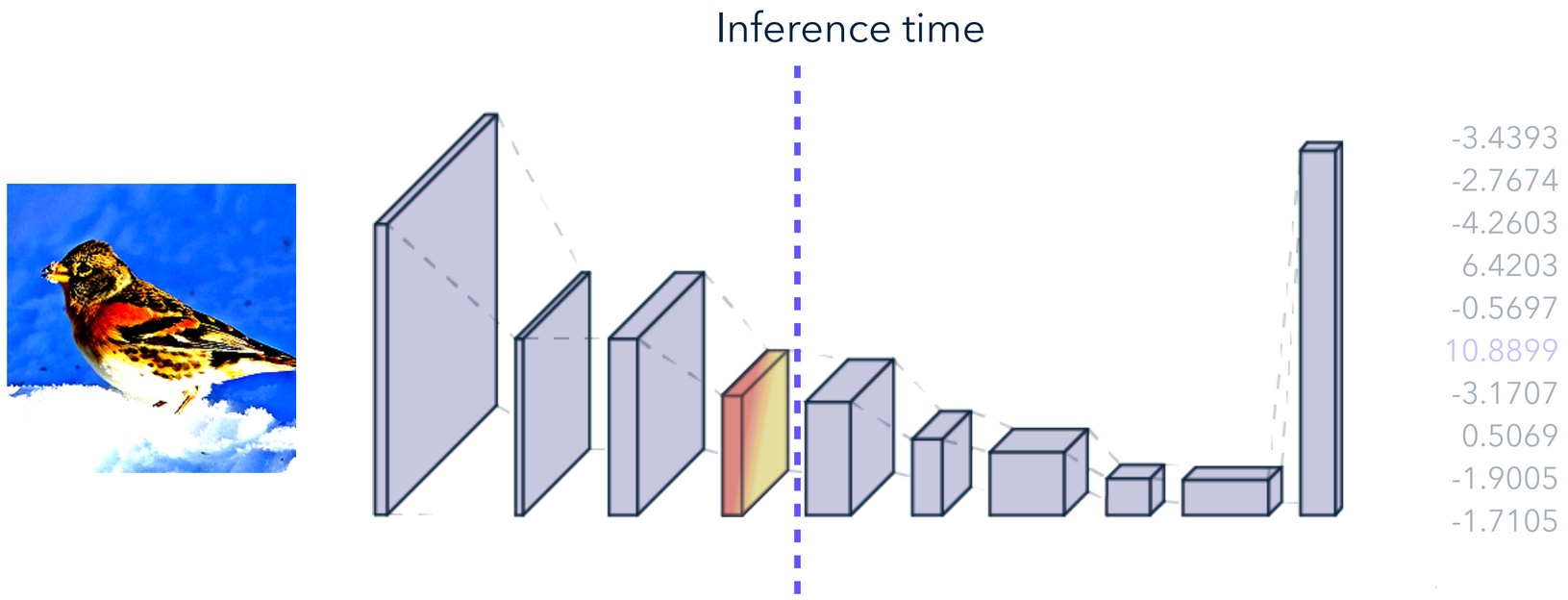


-3.4393
-2.7674
-4.2603
6.4203
-0.5697
10.8899
-3.1707
0.5069
-1.9005
-1.7105

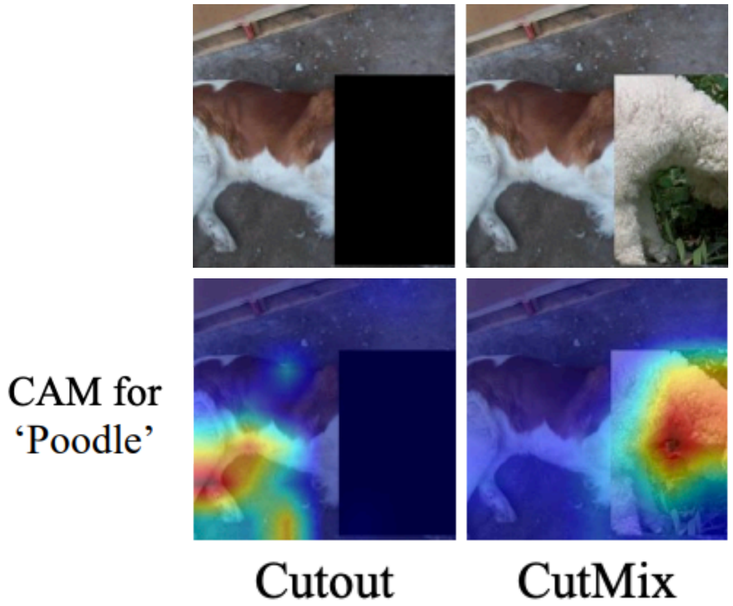
Saliency Map - CIFAR10 Example



Saliency Map - CIFAR10 Example

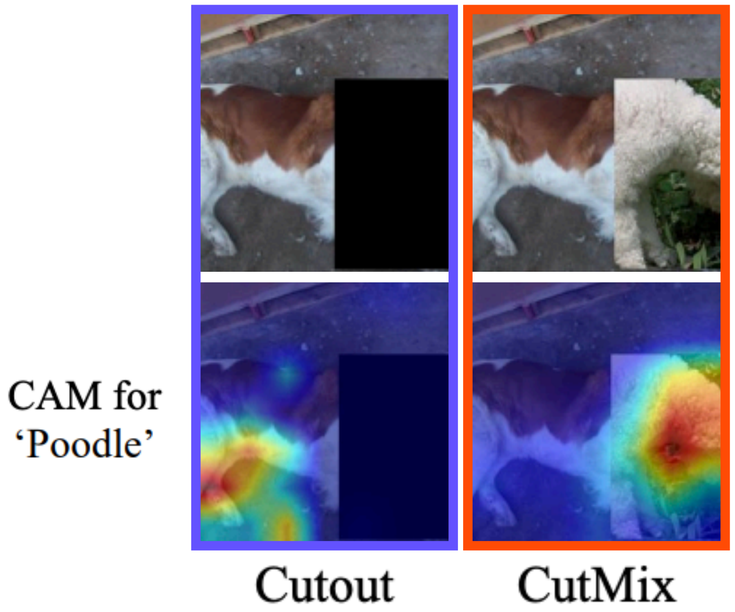


Saliency Map Comparison



Yun et al. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features

Saliency Map Comparison



Yun et al. (2019). CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features

Back to Basics: Derivative

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

- How much the output changes when I make an infinitesimally small change to the input

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

- How much the output changes when I make an infinitesimally small change to the input
- Robust feature - prediction doesn’t change when I make a perturbation

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

- How much the output changes when I make an infinitesimally small change to the input
- Robust feature - prediction doesn't change when I make a perturbation
- Non-robust feature - prediction is **sensitive** to perturbations

Back to Basics: Derivative

$$\lim_{h \to 0} \frac{f(x) - f(x+h)}{h}$$

- How much the output changes when I make an infinitesimally small change to the input
- Importance or sensitivity?

Saliency Map Comparison

- Looking at sensitivity could be insightful

Saliency Map Comparison

- Looking at sensitivity could be insightful

- But is it only capturing a small part of what a model learns

Saliency Map Comparison

- Looking at sensitivity could be insightful
- But is it only capturing a small part of what a model learns
- Sensitive features can be important, but not all important features are sensitive

Take-aways

- We don’t have good ways of interpreting what is going on inside DNNs

Take-aways

- We don’t have good ways of interpreting what is going on inside DNNs
- But we can use basic DL concepts to construct correct interpretations of existing techniques

Take-aways

- We don’t have good ways of interpreting what is going on inside DNNs
- But we can use basic DL concepts to construct correct interpretations of existing techniques
- And hopefully build better ones

Take-aways

- We don’t have good ways of interpreting what is going on inside DNNs
- But we can use basic DL concepts to construct correct interpretations of existing techniques
- And hopefully build better ones
- Understanding representations is very much work in progress