Learning De-biased Representations with Biased Representations

ICML'20

2020. 08. 06

Kakao Enterprise

Context Part

Presented by brian.me (Junsoo Lee)

Motivation

- Many machine learning algorithms are trained and evaluated by splitting data from a single source into training and test sets.
- Such *in-distribution* learning scenarios do not guarantee telling us if the models are relying on dataset biases as shortcuts for successful prediction, e.g., using snow cues for recognizing snowmobiles, resulting in biased models which fail to generalize when the bias shifts to a different classes, i.e., *cross-bias generalisation*.

Motivation

Neural networks tend to rely on texture bias as visual features.

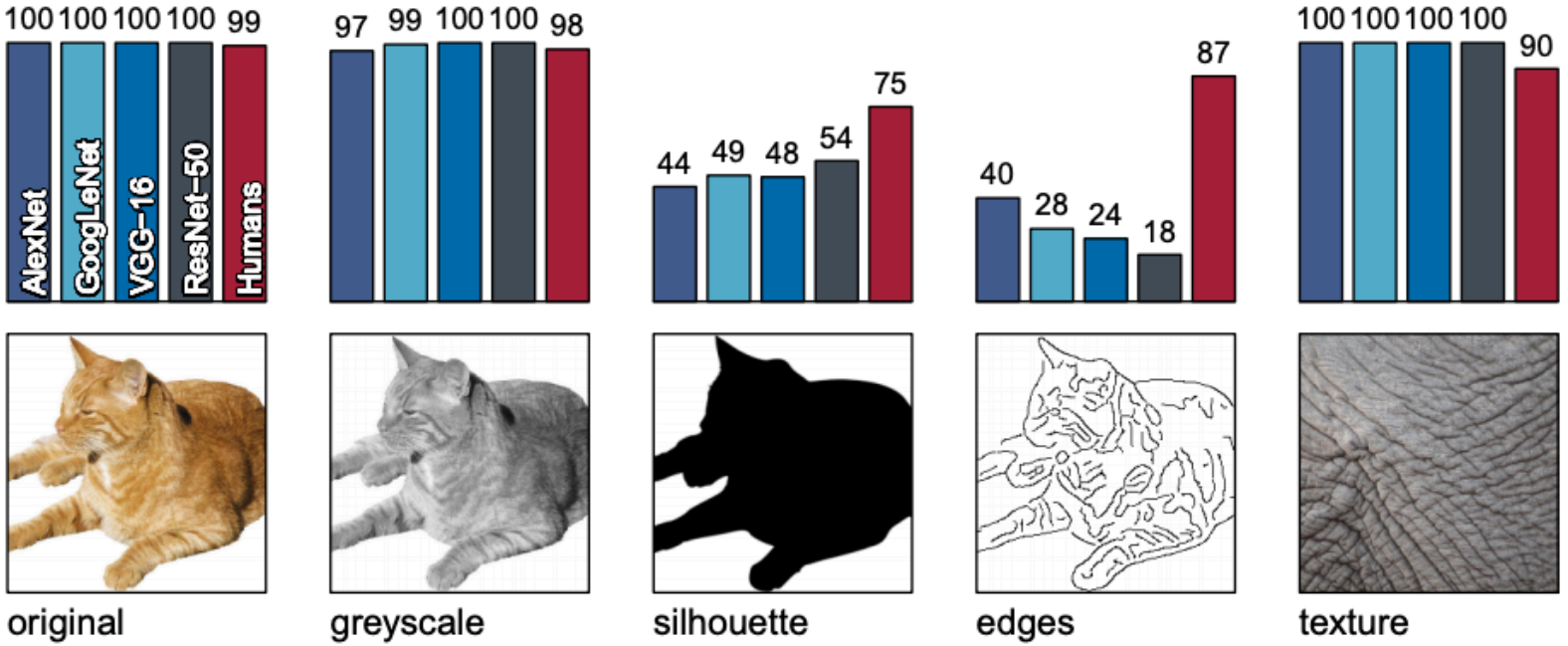


Figure 2: Accuracies and example stimuli for five different experiments without cue conflict.

[1] ImageNet-Trained CNNs are Biased towards Texture; Increasing Shape Bias Improves Accuracy and Robustness, ICLR'19

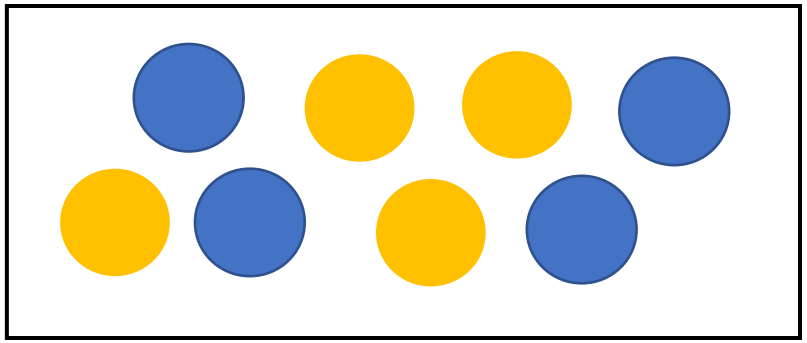
Why does this problem happen?

- This “cross-bias generalisation” problem is easy to happen because our model is good at finding much easier path, i.e., shortcuts, to solve the given problem.
- In other words, our model does not need to exploit its full capacity if it finds the “sufficiency” of bias cues for prediction of the target label in the training data.
- For example, predictions based on presence of certain words in language models without much reasoning, based on local texture in CNNs, based on static cues in action recognition models.

To remedy this problem,

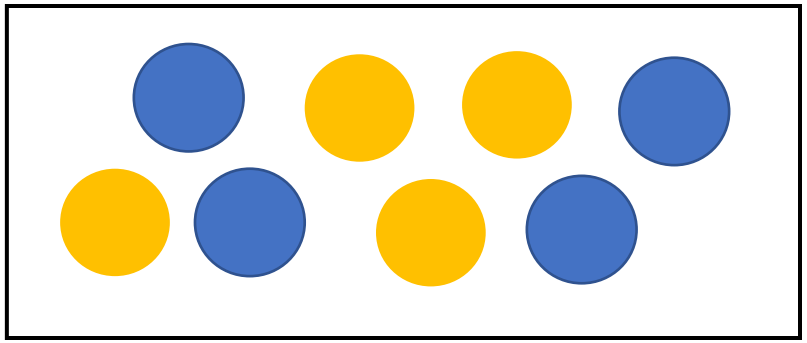
- This work proposes a training framework to train a de-biased representation by encouraging it to be **statistically independent** from representations obtained from an auxiliary networks that is designed to be strongly biased due to its architecture constraint.

Problem Definition



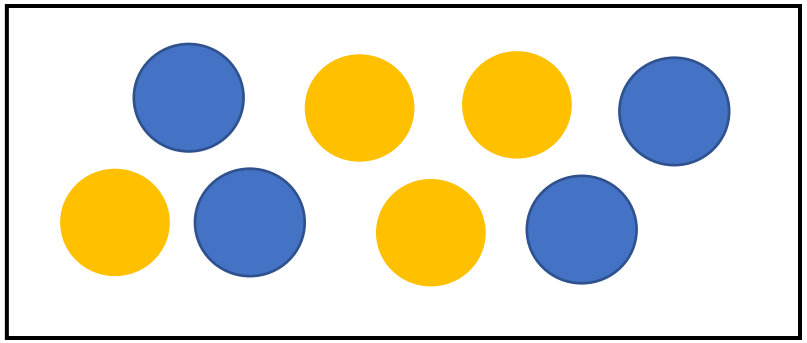
- We first define random variables, signal S and bias B as cues for the recognition of an input X as certain target variable Y.
- Signal S are the essential cues for recognition of X as Y.
- Biases B, on the other hand, are cues not essential but highly correlated with the target Y.

Problem Definition



- If $p(B|X)$ is fully known, we can directly encourage $f(X) \perp B$; however, these biases are not easily addressed by the handcraft approaches: (1) texture and shape cannot easily be disentangled, (2) collecting unusual images or building a generative model $p(X|B)$ is very expensive.

Proposed Method



- Instead of assuming explicit prior knowledge on $p(B|X)$, we can approximate B by defining a set of models G that are biased towards B by the architecture design.
- More precisely, we define $\boldsymbol{G}$ to be a **bias-characterising model** class.

Proposed Method

- We de-bias a representation f by designing a set of biased models G and letting f run away from G.
- This leads to the independence from bias cues B while leaving signal cues S as valid recognition cues.

Hilbert-Schmidt Independence Criterion (HSIC)

- TBD

Hilbert-Schmidt Independence Criterion (HSIC). Since we need to measure the degree of independence between continuous random variables $f(X)$ and $g(X)$ in high-dimensional spaces, it is infeasible to resort to histogram-based measures; we use HSIC (Gretton et al., 2005). For two random variables U and V and kernels k and l, HSIC is defined as $\text{HSIC}^{k,l}(U,V) := ||C^{k,l}_{UV}||^2_{\text{HS}}$ where $C^{k,l}$ is the cross-covariance operator in the Reproducing Kernel Hilbert Spaces (RKHS) of k and l (Gretton et al., 2005), an RKHS analogue of covariance matrices. $||\cdot||_{\text{HS}}$ is the Hilbert-Schmidt norm, a Hilbert-space analogue of the Frobenius norm. It is known that for two random variables U and V and radial basis function (RBF) kernels k and l, $\text{HSIC}^{k,l}(U,V) = 0$ if and only if $U \perp\!\!\!\perp V$. A finite-sample estimate of $\text{HSIC}^{k,l}(U,V)$ has been used in practice for statistical testing (Gretton et al., 2005; 2008), feature similarity measurement (Kornblith et al., 2019), and model regularisation (Quadrianto et al., 2019; Zhang et al., 2018). We employ an unbiased estimator $\text{HSIC}^{k,l}_1(U,V)$ (Song et al., 2012) with m samples, defined as

$$\text{HSIC}^{k,l}_1(U,V) = \frac{1}{m(m-3)}\left[\text{tr}(\widetilde{U}\widetilde{V}^T) + \frac{\mathbf{1}^T\widetilde{U}\mathbf{1}\mathbf{1}^T\widetilde{V}^T\mathbf{1}}{(m-1)(m-2)} - \frac{2}{m-2}\mathbf{1}^T\widetilde{U}\widetilde{V}^T\mathbf{1}\right]$$

where $\widetilde{U}_{ij} = (1-\delta_{ij})\,k(u_i,u_j)$, $\{u_i\} \sim U$, *i.e.*, the diagonal entries of $\widetilde{U}$ are set to zero. $\widetilde{V}$ is defined similarly.

Full Objective

$$\min_f \left\{ \mathcal{L}(f) + \lambda \max_g \Big(\mathbf{HSIC}_1(f, g) - \lambda_g \mathcal{L}(g) \Big) \right\}. \quad (3)$$

- Having specified G to represent the bias B, we need to train $g \in G$ for the original task to intentionally overfit G to B.
- Thus, the inner optimisation involves both the independence criterion and the original task loss $L(g)$.
- Eq (3) is solved in alternative update.

Exp (1) : Biased MNIST

Figure 2. **Biased MNIST.** A synthetic dataset with the colour bias which highly correlates with the labels during training.

- Lets verify this framework on datasets where we have full control over the type and amount of bias during training and evaluation, i.e., toy setting.
-

Exp (1) : Biased MNIST

Figure 2. **Biased MNIST.** A synthetic dataset with the colour bias which highly correlates with the labels during training.

	Biased						Unbiased					
ρ	Vanilla	Biased	HEX	LearnedMixin	RUBi	ReBias (ours)	Vanilla	Biased	HEX	LearnedMixin	RUBi	ReBias (ours)
.999	**100.**	**100.**	71.3	2.9	99.9	**100.**	10.4	10.	10.8	12.1	13.7	**22.7**
.997	**100.**	**100.**	77.7	6.7	99.4	**100.**	33.4	10.	16.6	50.2	43.0	**64.2**
.995	**100.**	**100.**	80.8	17.5	99.5	**100.**	72.1	10.	19.7	78.2	**90.4**	76.0
.990	**100.**	**100.**	66.6	33.6	**100.**	**100.**	89.1	10.	24.7	88.3	**93.6**	88.1
avg.	**100.**	**100.**	74.1	15.2	99.7	**100.**	51.2	10.	18.0	57.2	60.2	**62.7**

Table 1. **Biased MNIST results.** Biased and unbiased accuracies on varying train correlation ρ. Besides our results, we report vanilla F, G and previous methods. Ours are shown in gray columns. Each value is the average of three different runs.

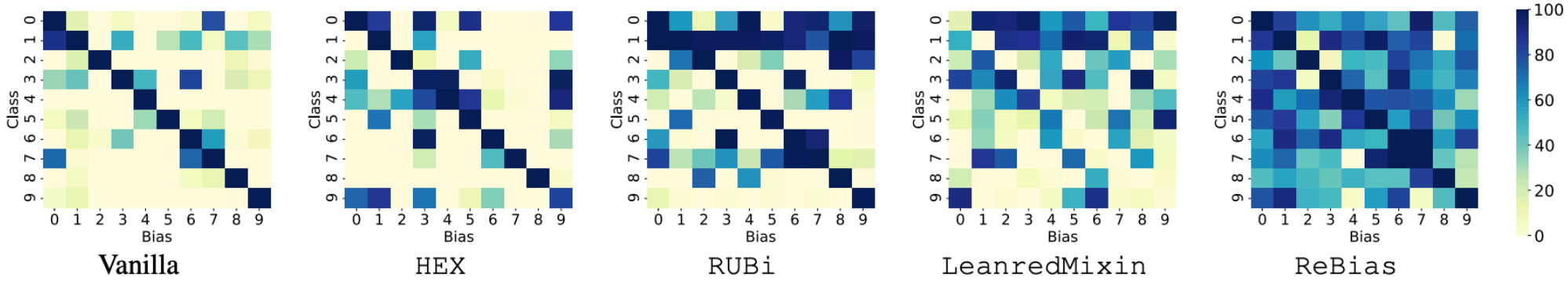


Figure 3. **Accuracy per bias-class pair.** We show accuracies for each bias and class pair $(B, Y) = (b, y)$ on Biased MNIST. All methods are trained with $\rho = 0.997$. The diagonals in each matrix indicate the pre-defined bias-target pair (§4.2.1). The number of samples per (b, y) cell is identical across all pairs (unbiased test set).

Exp (1) : Biased MNIST

Figure 2. **Biased MNIST.** A synthetic dataset with the colour bias which highly correlates with the labels during training.

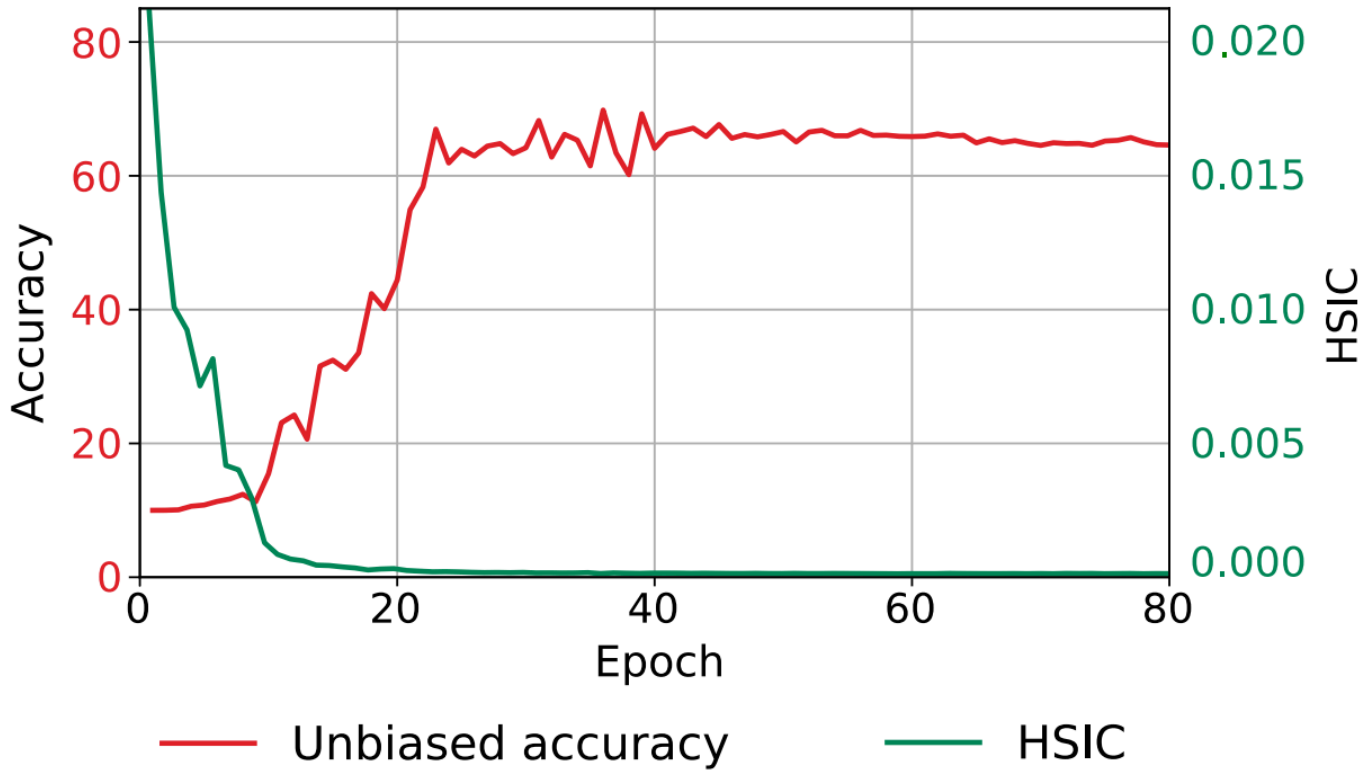
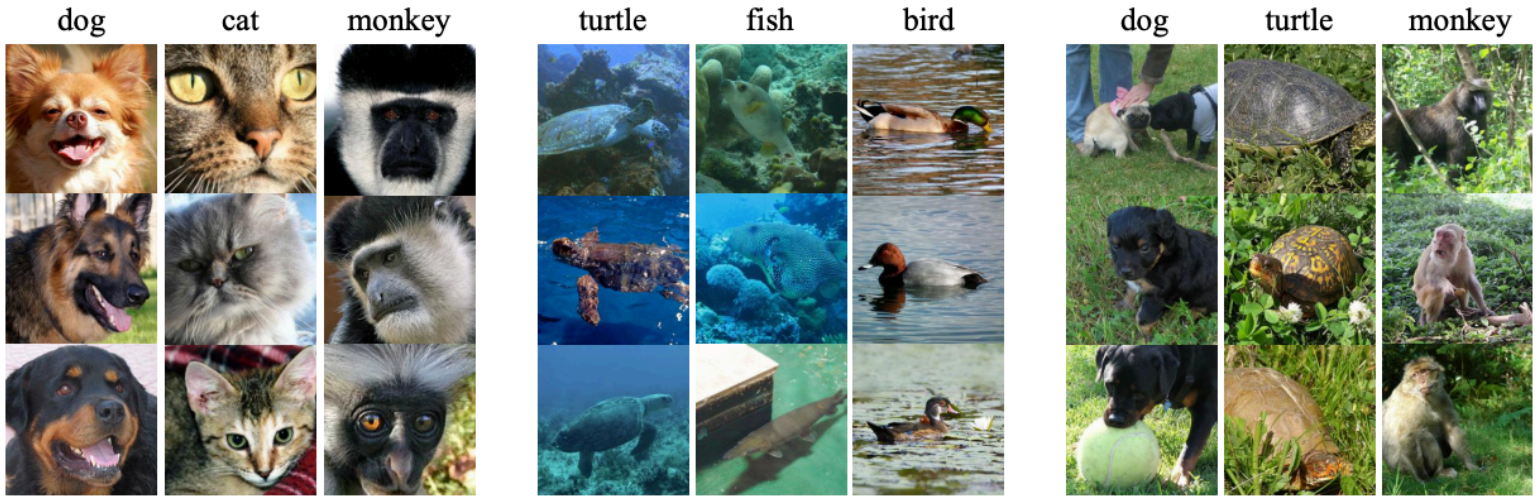


Figure 4. **Learning curves.** `ReBias` achieves better generalisation by minimizing HSIC between representations.

Exp (2) : Real Domain (ImageNet)

clusters. Example images from texture clusters. Each cluster is named according to the con



(a) "Face" cluster samples (b) "Water" cluster samples (c) "Grass" cluster samples

Figure A3. **Dominant texture-class correlations.** For each cluster, we visualise its top-3 correlated classes.

- Since it is difficult to evaluate the cross-bias generalisability on realistic data, a surrogate measurement is introduced.
- A combination-wise accuracy $A_{c,y}$ is computed by $\text{Corr}(c, y)/\text{Pop}(c, y)$, where Corr is the number of correctly predicted samples and Pop is the total number of samples.

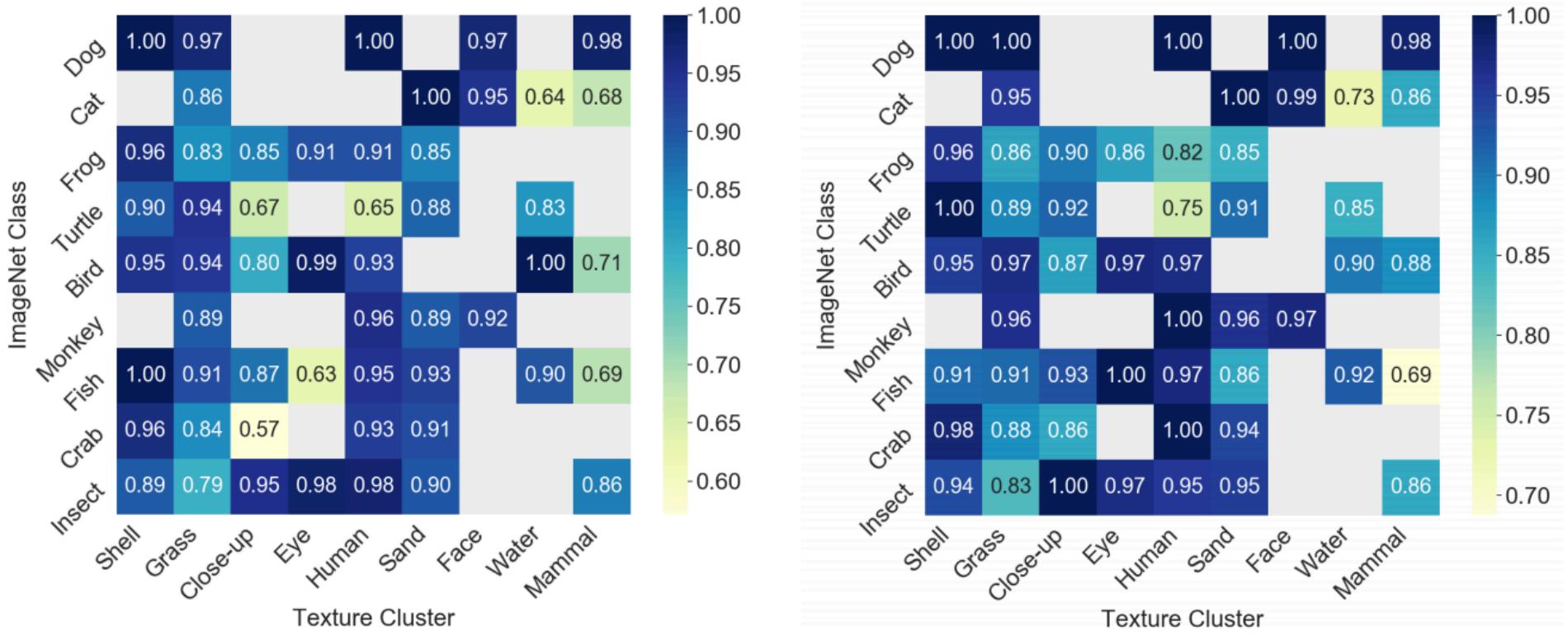
Exp (2) : Real Domain (ImageNet)

Model description	Biased	Unbiased	IN-A	IN-C
Vanilla (ResNet18)	90.8	88.8	24.9	54.2
Biased (BagNet18)	67.7	65.9	18.8	31.7
StylisedIN (Geirhos et al., 2019)	88.4	86.6	24.6	**61.1**
LearnedMixin (Clark et al., 2019)	64.1	62.7	15.0	27.5
RUBi (Cadene et al., 2019)	90.5	88.6	27.7	53.7
ReBias (ours)	**91.9**	**90.5**	**29.6**	57.5

Table 2. **ImageNet results.** We show results corresponding to F =ResNet18 and G =BagNet18. IN-A and IN-C indicates ImageNet-A and ImageNet-C, respectively. We repeat each experiment three times.

- ImageNet-A: contains the failure cases of resnet50 among web images. The images usually consist of frequently appearing background elements, which become erroneous cues for recognition.
- ImageNet-C: is proposed to evaluate robustness to 15 corruption types including “noise”, “blur”, “weather”, …

Exp (2) : Real Domain (ImageNet)



(a) Vanilla trained `ResNet18`.

(b) `ReBias` trained `ResNet18`.

igure A4. **Texture-class-wise accuracies on ImageNet.** For every texture-class pair $(B, Y) = (b, y)$, corresponding accuracy is isualised. We ignore cells with population less than 10 (masked in gray).

Exp (3) : Real Domain (Action Recognition)

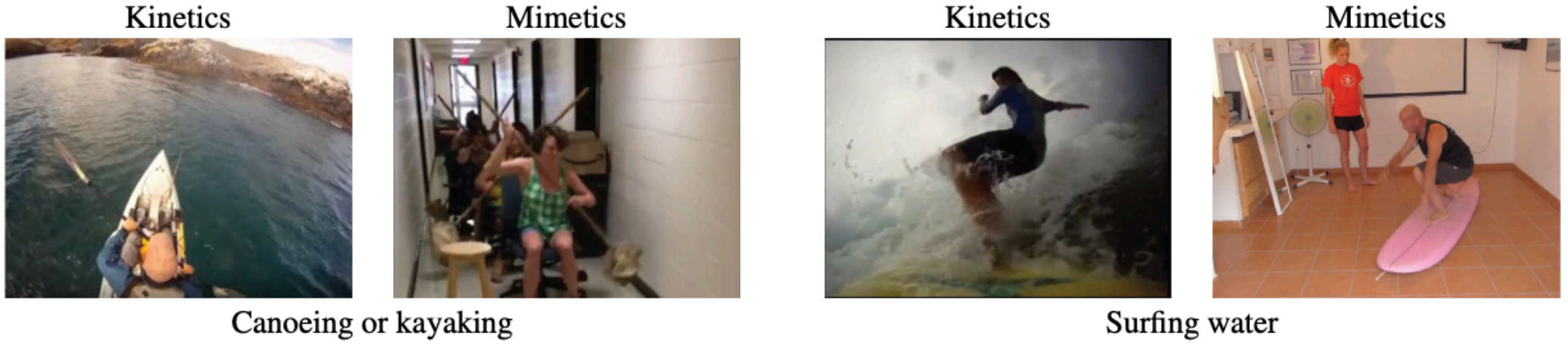


Figure A5. **Kinetics and Mimetics datasets.** We show examples of two classes, "canoeing or kayaking" and "surfing water". Kinetics samples are biased towards certain scene contexts (*e.g.* ocean), but Mimetics is relatively free from such context biases.

- While Kinetics samples are biased towards static cues like scene and objects, Mimetics are relatively free of such correlations. Mimetics is thus a suitable benchmark for validating the cross-bias generalisation performances.
- F: 3D-ResNet18, G: 2D-ResNet18

Exp (3) : Real Domain (Action Recognition)

Model description	Biased (Kinetics)	Unbiased (Mimetics)
Vanilla (3D-ResNet18)	54.5	18.9
Biased (2D-ResNet18)	50.7	18.4
LearnedMixin (Clark et al., 2019)	12.3	11.4
RUBi (Cadene et al., 2019)	22.4	13.4
ReBias (ours)	**55.8**	**22.4**

Table 3. **Action recognition results.** We show results corresponding to F =3D-ResNet18 and G =2D-ResNet18 with baseline comparisons. Top-1 accuracies are reported. Each result is the average of three runs.

Summary

- This work have identified very practical problem faced by many machine learning algorithms that the learned models exploit bias shortcuts to recognise the target, i.e., cross-bias generalisation problem.
- Given an identified set of models G that encodes the bias to be removed, this framework encourages a model f to be statistically independent of G.

+Discusstion

- Is HSIC an unique measurement for computing statistical independence between two representations?
  - Beta-VAE, Factor-VAE, Cascade-VAE …?
- Can we adopt the main idea of this work into my topic, unsupervised / semi-supervised representation learning?
  - Signals/Bias <-> Class-shared / Instance-specific Information