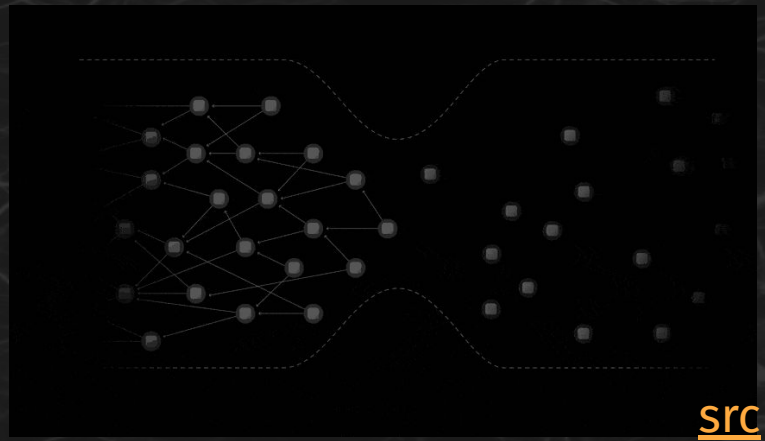
The Distributed Information Bottleneck Reveals the Explanatory Structure of Complex Systems

Kieran A. Murphy and Dani S. Bassett



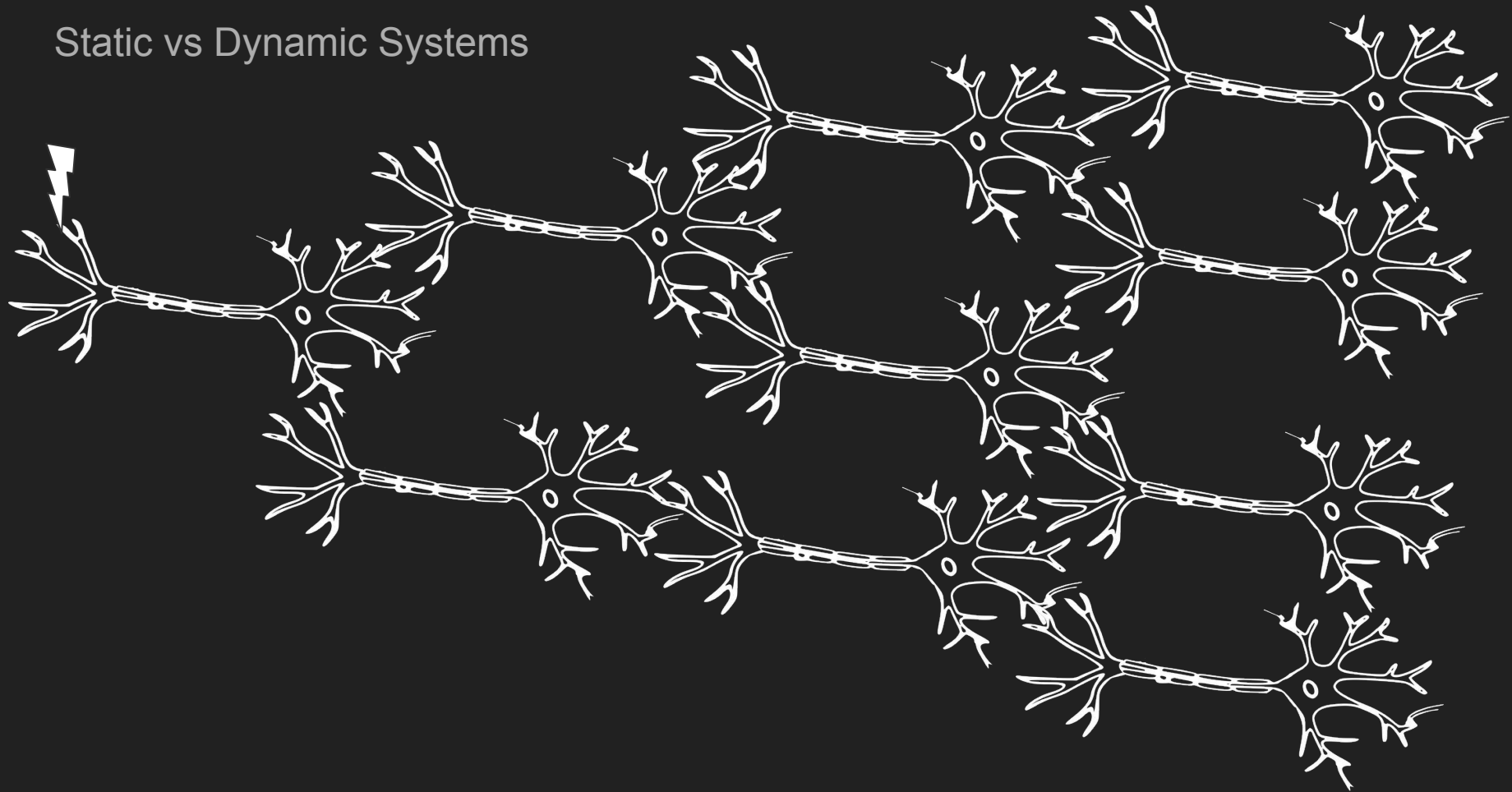
src

Motivation

src

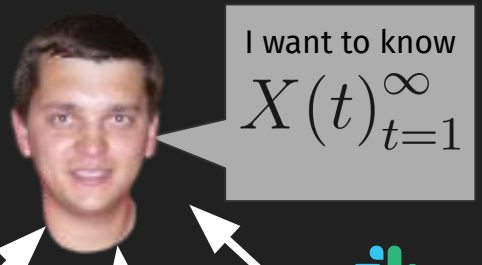
Complex Systems and How We Interpret Them

Static vs Dynamic Systems



“The Postdoc Problem” - Multiterminal Source Coding

- No communication between postdoc or from mentor
- Postdocs have unique distorted signal
- Mentor aggregates from each postdoc
- Rate limited comms



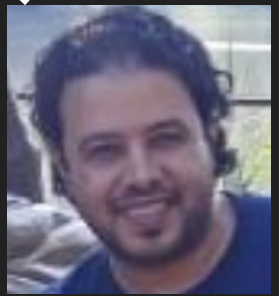
SMPD = Slack Messages per Day

1 Million SMPD

10 SMPD

20 SMPD

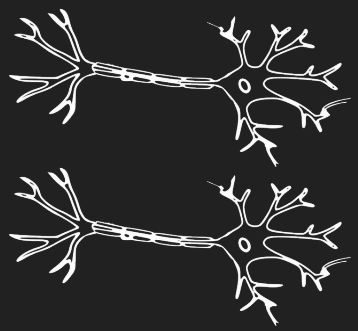
30 SMPD

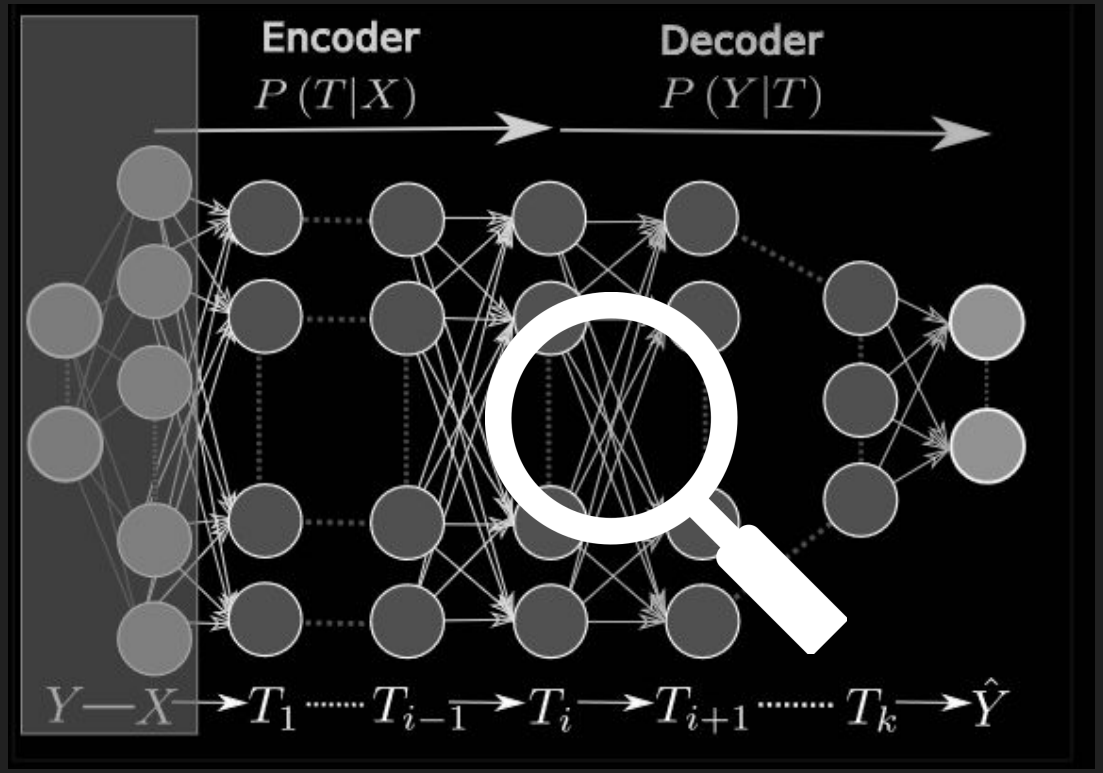


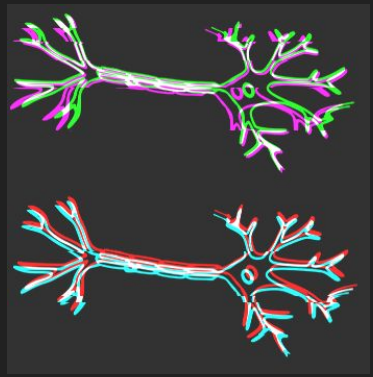
Distortion rate of knowing X is asymptotically small with infinite postdocs

System Interpretability via Deep Learning

Using interpretability methods to understand complex systems







Background

src

Probability Refresher

Continuous:

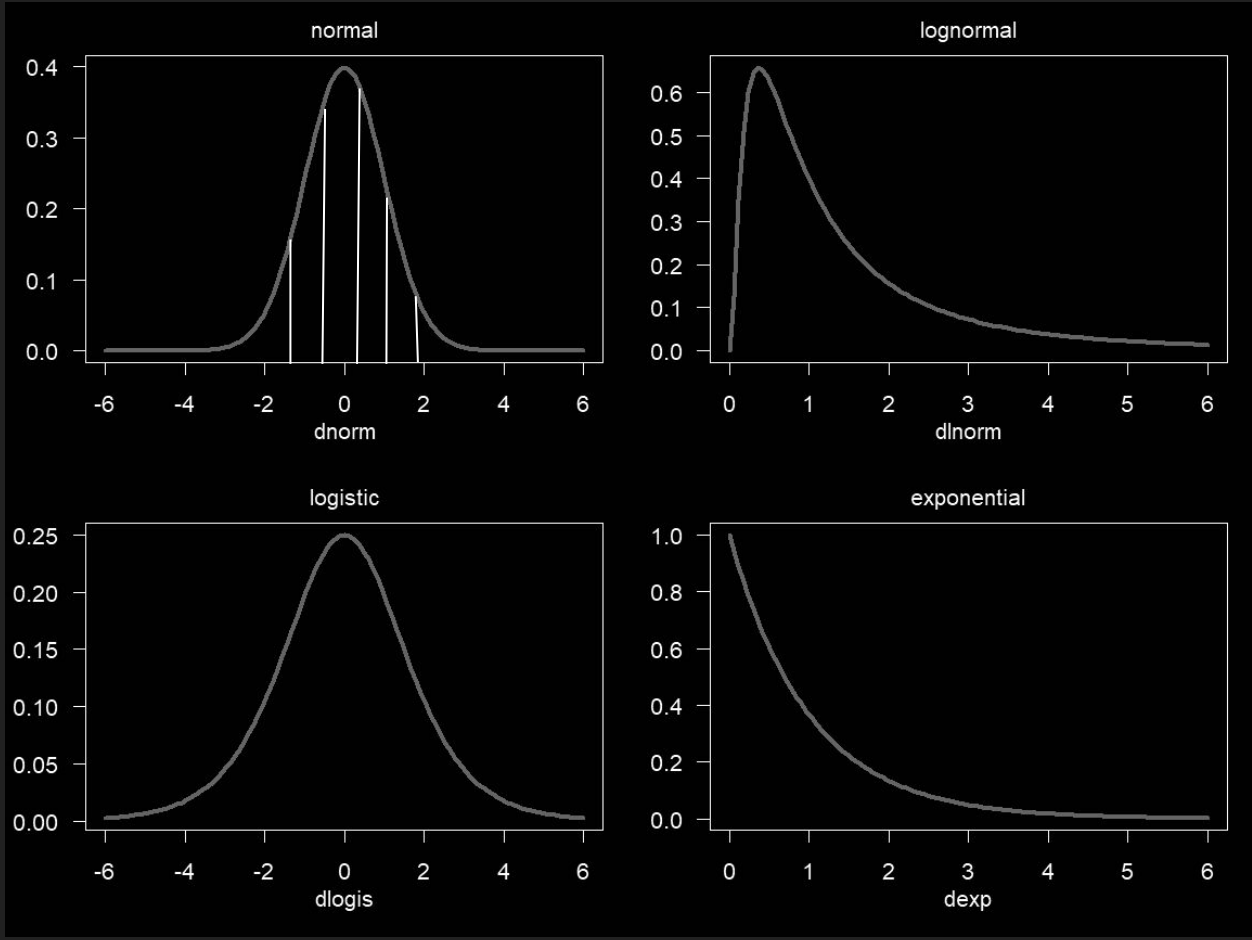
Probability Density Function

$$\text{PR}[a \leq X \leq b] = \int_a^b f_X(x)dx$$

Discrete:

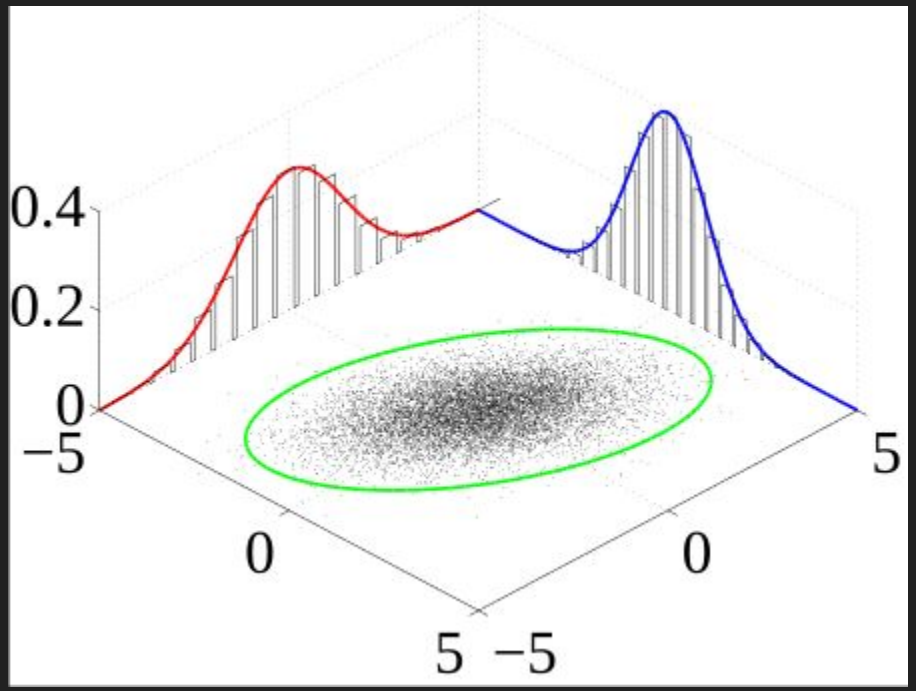
Probability Mass Function

$$1 = \sum_x p_X(x)$$



Joint and Conditional Probabilities, Chain Rule, and Bayes Theorem

For random variables A, B the conditional probabilities, $P(A|B), P(B|A)$, and joint probability $P(A, B)$ have the following identities



Definition of Conditional Probability

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

Chain Rule

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A)$$

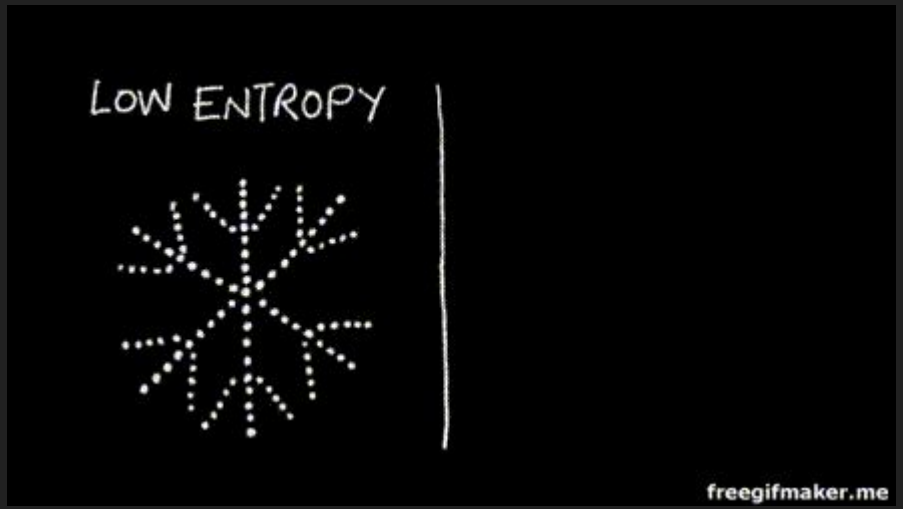
Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Shannon Entropy

For discrete random variable X with PMF $p_X(x)$

$$-\sum p_X(x) \log_2 p(x)$$



Two fair coin tosses

Intuitively - Shannon Entropy measures “uncertainty” or “surprise”

Joint + Conditional Shannon Entropy (Variants)

For discrete random variables X and Y with joint and conditional PMFs $p(X, Y)$ $p(X|Y)$

Joint Entropy

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x, y)$$

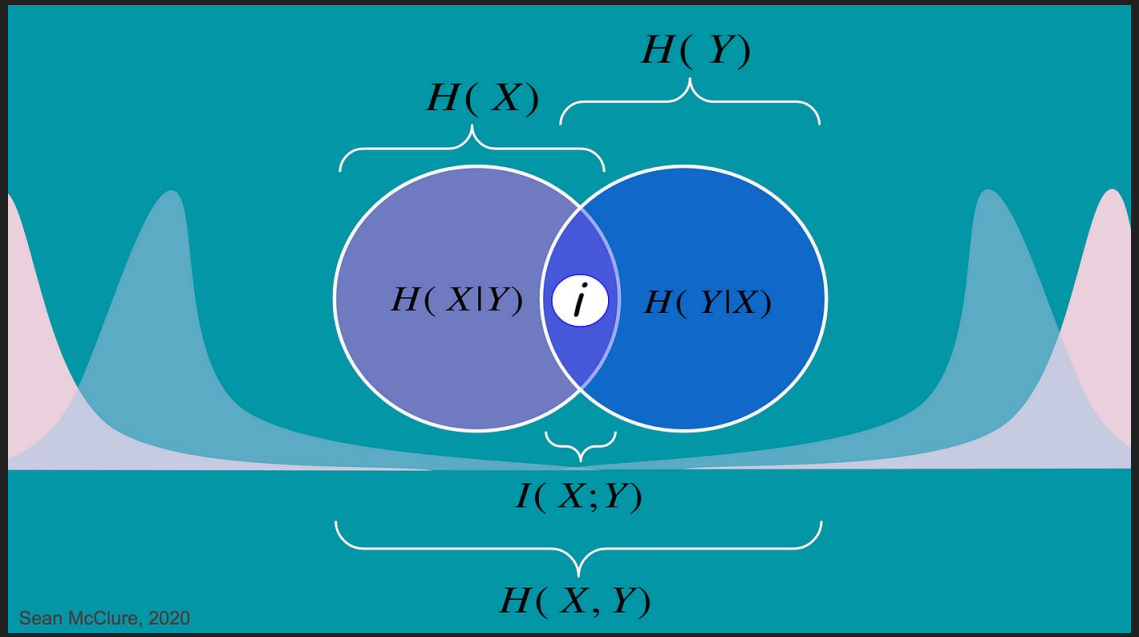
Conditional Entropy

$$H(Y|X) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 \frac{p(x, y)}{p(x)}$$

Mutual Information

The reduction in uncertainty of one variable due to presence of another variable, defined as

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

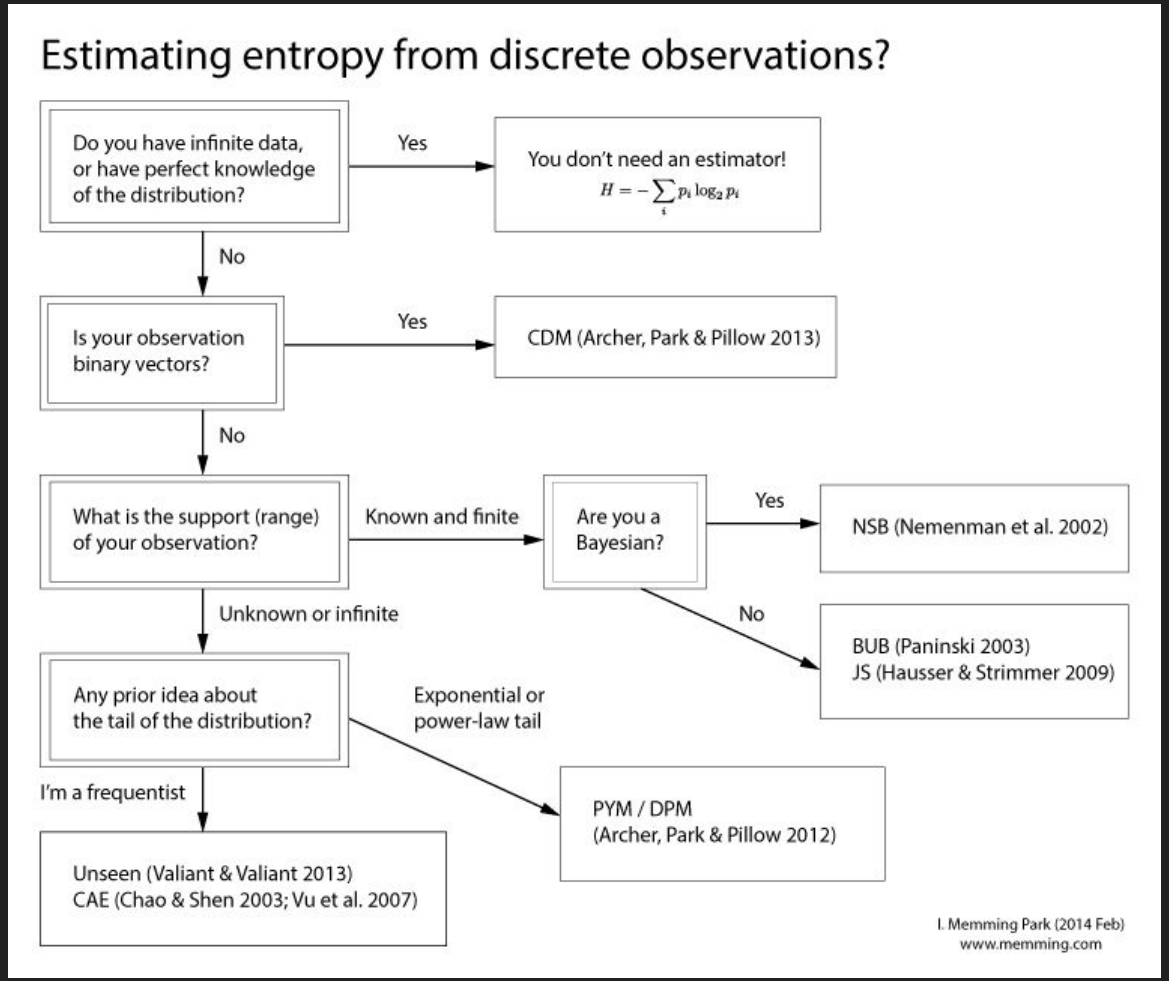


Information Theoretic Estimation

Basically, if you don't know the distribution perfectly

NO estimator is perfect

Most use histogram-based estimation



The Information Bottleneck

What is the information bottleneck intuitively?

Imagine I have a long message, but have to send it over a compressed channel and reconstruct it

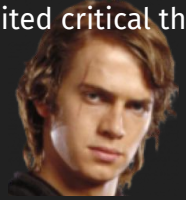
Did you ever hear the tragedy of Darth Plagueis The Wise? I thought not.

It's not a story the Jedi would tell you. It's a Sith legend. Darth Plagueis was a Dark Lord of the Sith, so powerful and so wise he could use the Force to influence the midichlorians to create life...

He had such a knowledge of the dark side that he could even keep the ones he cared about from dying. The dark side of the Force is a pathway to many abilities some consider to be unnatural. He became so powerful... the only thing he was afraid of was losing his power, which eventually, of course, he did.

Unfortunately, he taught his apprentice everything he knew, then his apprentice killed him in his sleep. Ironic. He could save others from death, but not himself.

Bottleneck
(limited critical thinking)



Murder is fine probably

What is the information bottleneck mathematically?

Recall

$$I(X;Y) = H(X) - H(X|Y)$$

The idea is we have a constrained representation $U = f(X)$ and choose a scalar parameter β to control bottleneck strength giving. See [Tishby 2000]

$$\mathcal{L}_{\text{IB}} = \beta I(U;X) - I(U;Y)$$

Aside: the Variational Information Bottleneck

Because of difficulty with MI estimation, current approaches use a variational estimator from this paper: Deep Variational Information Bottleneck which measures the predictability of Y given X using the KL divergence:

$$\mathcal{L}_{\text{VIB}} = \beta D_{\text{KL}}(p(u|x)||r(u)) - \mathbb{E}[\log p(y|u)]$$

The Information Bottleneck in Deep Learning

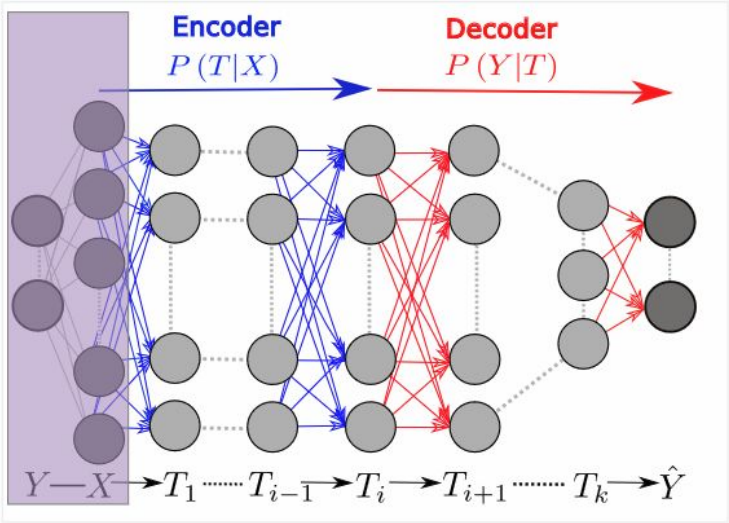


Figure 1: The DNN layers form a Markov chain of successive internal representations of the input layer X. Any representation of the input, T, is defined through an encoder, $P(T|X)$, and a decoder $P(\hat{Y}|T)$, and can be quantified by its *information plane* coordinates: $I_X = I(X;T)$ and $I_Y = I(T;Y)$. The Information Bottleneck bound characterizes the optimal representations, which maximally compress the input X, for a given mutual information on the desired output Y. After training, the network receives an input X, and successively processes it through the layers, which form a Markov chain, to the predicted output $\hat{Y}$. $I(Y;\hat{Y})/I(X;Y)$ quantifies how much of the relevant information is captured by the network.

See

Shwartz-Ziv, Tishby et al. 2017

And

Tishby et al. 2015

The Information Plane (Shwartz-Ziv et al. 2017)

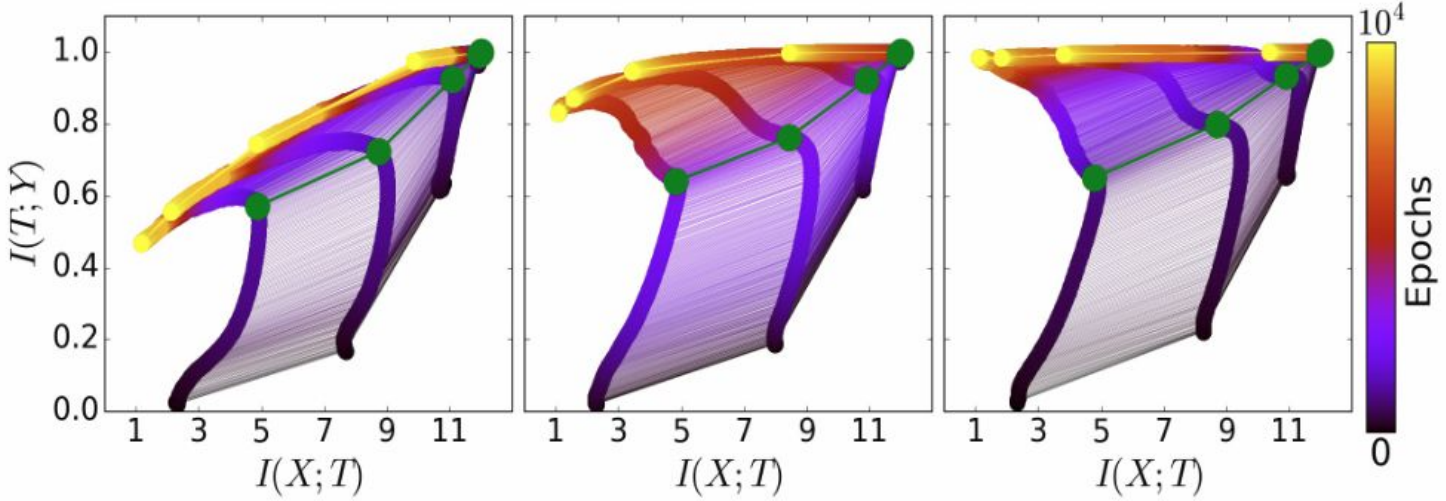


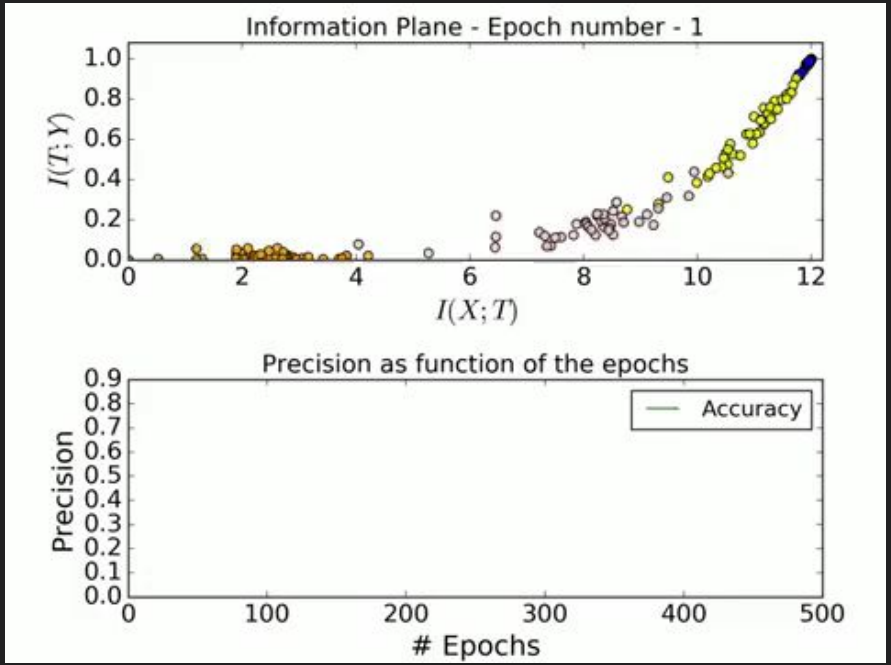
Figure 3: The evolution of the layers with the training epochs in the information plane, for different training samples. On the left - 5% of the data, middle - 45% of the data, and right - 85% of the data. The colors indicate the number of training epochs with Stochastic Gradient Descent from 0 to 10000. The network architecture was fully connected layers, with widths: input=12-10-8-6-4-2-1=output. The examples were generated by the spherical symmetric rule described in the text. The green paths correspond to the SGD drift-diffusion phase transition - grey line on Figure 4

The Shwartz-Ziv/Tishby Explanation of Training Dynamics

Two phases during training:

1. Increase in information around labels with increase in information around features
2. Decrease in information Around features with increase In label information. “Compression” phase

Claim the second phase leads To better generalization



The Saxe Rebuttal: Well... it depends (Saxe et al. 2018)

Essentially:

- It depends on nonlinearities
- RELU, Linear and TanH nonlinearities all behave differently
- Also there is no relationship between “compression” and generalization ability
- See [Saxe et al. 2018]

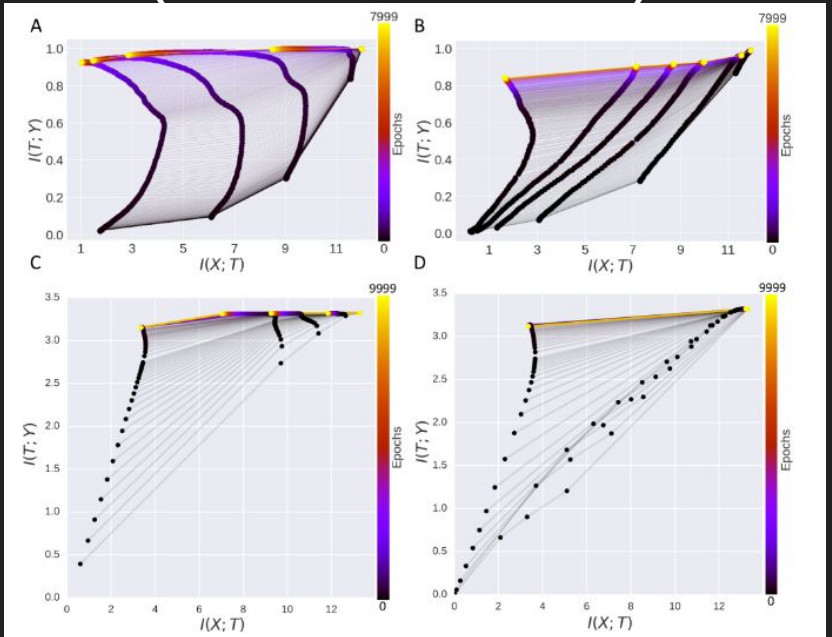


Figure 1: Information plane dynamics and neural nonlinearities. (A) Replication of Shwartz-Ziv & Tishby (2017) for a network with tanh nonlinearities (except for the final classification layer which contains two sigmoidal neurons). The x-axis plots information between each layer and the input, while the y-axis plots information between each layer and the output. The color scale indicates training time in epochs. Each of the six layers produces a curve in the information plane with the input layer at far right, output layer at the far left. Different layers at the same epoch are connected by fine lines. (B) Information plane dynamics with ReLU nonlinearities (except for the final layer of 2 sigmoidal neurons). Here no compression phase is visible in the ReLU layers. For learning curves of both networks, see Appendix A. (C) Information plane dynamics for a tanh network of size $784 - 1024 - 20 - 20 - 20 - 10$ trained on MNIST, estimated using the non-parametric kernel density mutual information estimator of Kolchinsky & Tracey (2017); Kolchinsky et al. (2017), no compression is observed except in the final classification layer with sigmoidal neurons. See Appendix B for the KDE MI method applied to the original Tishby dataset; additional results using a second popular nonparametric k-NN-based method (Kraskov et al., 2004); and results for other neural nonlinearities.

The Drama (A remind… …n read OpenReview)

Essentiall… …aims of this ICLR submission all together.

Readers: Everyone Show Revisions

ICLR 2018 Conference Acceptance Decision

ICLR 2018 Conference Program Chairs

29 Jan 2018, 13:14 (modified: 29 Jan 2018, 16:08) ICLR 2018 Conference Acceptance Decision Readers: Everyone Show Revisions

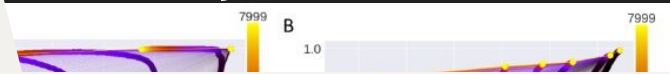
Decision: Accept (Poster)

Comment:

This submission explores recent theoretical work by Shwartz-Ziv and Tishby on explaining the generalization ability of deep networks. The paper gives counter-examples that suggest aspects of the theory might not be relevant for all neural networks.

There is some uncertainty surrounding the results where mutual information is estimated empirically. Even state-of-the-art estimation methods might lead to misleading empirical results. However, the submission appears to follow reasonable practice following previous work, making the reported results at least suggestive. They warrant reporting for further study and discussion.

The reviewers generally found the paper interesting enough for acceptance, however strong objections were posted by Tishby. A lengthy public exchange resulted between the groups of authors. Not every part of this exchange is resolved. It is not clear whether Tishby's group would be able to fix the full-connected ReLU demonstration in this paper, or whether the authors of this submission have anything to say about Tishby's ReLU+convnet demonstration. By accepting this work, we are not declaring where this debate will end. However, we felt the current submission is a constructive part of ongoing discussion in the literature on furthering our theoretical understanding of neural networks.



…that there is no evident causal connection between compression and generalization" We rigorously pro… improvement in generalization, providing that the partitions remained homogenous to the label probability… representation compression (under these conditions) is effective as doubling the size of the training data! H… our presentations: Figure 2 (see attachment).

We note that the caveat is critical ("providing that the partitions remained homogenous to the label probabili… in which all inputs associated with the same discrete value have the same class label would be of benefit if po… given for why deep networks might achieve this special label-homogeneous partition if they in fact do, and this would seem to be the core fact to be explained. We also note that we observe similar generalization performance between Tanh and ReLU networks despite … dynamics, indicating that compression is not a major factor in the empirical behavior we ob… weak).

…the linear stu… there is no noise in the training dynamics, no Gibbs distribution on the weights, and yet nevertheless we observe nearly identical dynamics in the information plane.

Using the simple three neuron model, we show clearly that nonlinearity and the binning procedure can cause compression in this instance. This is our main point, which addresses a core claim of the information bottleneck theory of deep learning: compression does not appear to happen through a stochastic relaxation because (a) the randomness in SGD does not behave like a diffusion, (b) we observe identical compression even with true batch GD, where there is no noise and no stochastic relaxation, and (c) we have identified a simple mechanism that explains the observed empirical results based on the neural nonlinearity. We disagree with the statements "the diffusion phase mostly adds random noise to the weights, and they evolve like Wiener processes..." and "The stochasticity of SGD methods is usually motivated as a way of escaping local minima of the training error. In this paper we give it a new, perhaps much more important role: it generates highly efficient internal representations through compression by diffusion" for the reasons outlined above.

…no rigorous argument is

…e far left. Different layers at the same epoch are connected …amics with ReLU nonlinearities (except for the final layer …ression phase is visible in the ReLU layers. For learning …A. (C) Information plane dynamics for a tanh network of …ained on MNIST, estimated using the non-parametric kernel …f Kolchinsky & Tracey (2017); Kolchinsky et al. (2017), …he final classification layer with sigmoidal neurons. See …plied to the original Tishby dataset; additional results using …based method (Kraskov et al., 2004); and results for other neural nonlinearities.

The Distributed Information Bottleneck

Multiple Components

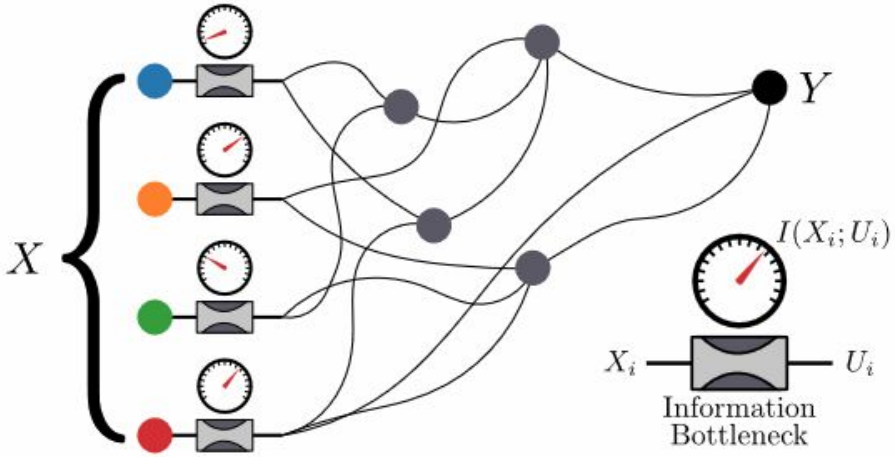


FIG. 1. **Distributed Information Bottleneck for insight into complex relationships.** Here, an input X has multiple components $\{X_i\}$ that share some amount of information with an output Y. The grey nodes represent interaction terms between the components, and the connections between nodes indicate information flow. We show in this work that distributing bottlenecks on the information from different components of the input X throttles the downstream complexity of the interactions and yields a continuum of approximations of the relationship between X and Y. The amount of information passing through each bottleneck—from each X_i into a learned representation U_i—reflects the relevance of each component for predicting Y, for each level of approximation.

Breaking Down the DIB Equation

$$\mathcal{L}_{\text{DIB}} = \beta \sum_i I(U_i; X_i) - I(U_X; Y)$$

Making it Variational with the InfoNCE Loss

When Y is continuous, the VIB is often estimated just by discretizing the support and treating the problem as classification. This requires high-resolution discretization for reasonable results typically, and is not practical. So they use the InfoNCE loss

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{i}^{n} \log \frac{\exp(s(u_X^{(i)}, u_Y^{(i)}/\tau))}{\exp(\sum_j^n s(u_X^{(i)}, u_Y^{(j)}/\tau))}$$

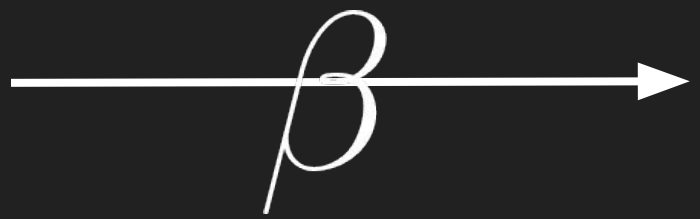
Where s is a similarity measure (e.g. Euclidean distance) and tau acts as an effective temperature. This is very similar to standard cross-entropy loss.

The Distributed Information Bottleneck Results

Sweeping Through the Bottleneck

Small beta

β



Open bottleneck

Large beta

Tight bottleneck

Black Box Boolean Circuits

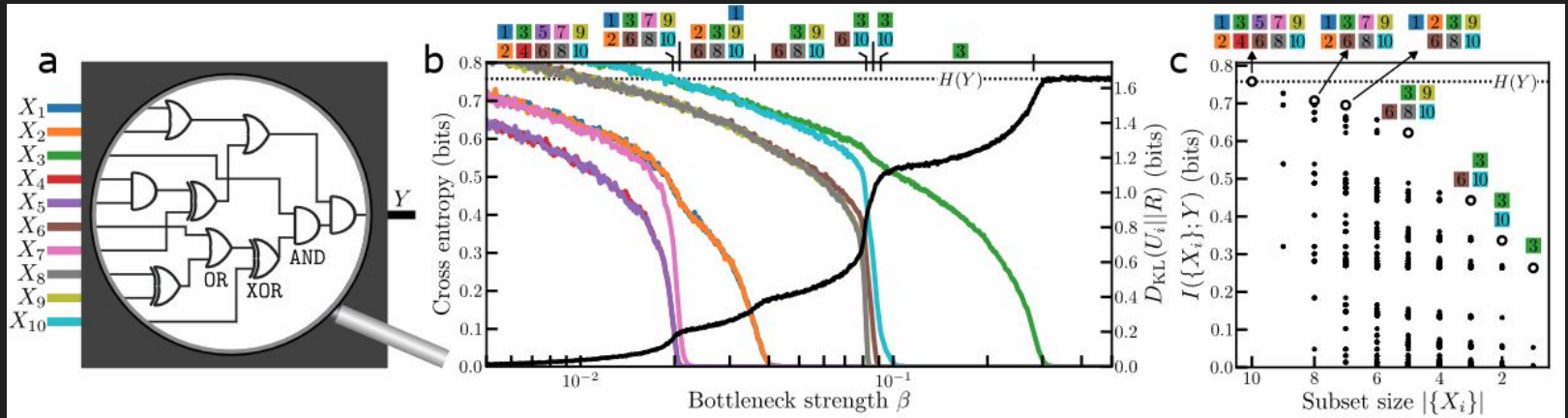


FIG. 2. **Opening a black-box Boolean circuit with the Distributed Information Bottleneck.** **(a)** A Boolean circuit has ten binary inputs $\{X_i\}$ connected through `AND`, `OR`, and `XOR` gates to one binary output Y. **(b)** With the Distributed IB, each input is compressed and the training objective (Eqn. 4) balances predictability of Y with the sum total of information conveyed about each input. Sweeping over the bottleneck strength β finds a series of relationships between compressed input components and the output Y. The cross entropy error of each relationship's prediction of Y, shown in black (left vertical axis), is nearly zero when the bottleneck is weakly applied (small β) and obtains its maximum value, the entropy $H(Y)$ (dotted line), after $\beta \approx 0.3$. Information transmitted about each of the inputs (colors corresponding to input gates in panel **(a)**) is measured through the proxy quantity $D_{\mathrm{KL}}(U_i||R)$ (right vertical axis). Information about the X_i decreases heterogeneously as the bottleneck tightens, with more information allocated to the more relevant components for predicting Y. Over the course of the β sweep, the scheme of approximations of the relationship between X and Y utilizes different subsets of the inputs (those above a threshold $D_{\mathrm{KL}}(U_i||R)$ are indicated at the top of the plot). **(c)** The mutual information $I(\{X_i\};Y)$ between all subsets of input channels $\{X_i\}$ and the output Y are shown as black circles; there is a large range in the amount of information that different subsets contain with respect to Y. The maximum mutual information arises from the combination of all ten inputs and the output, equal to the entropy $H(Y)$ (dotted line). Every subset of inputs utilized by the Distributed IB in the approximation scheme in **(b)** is the subset with maximal information for its size (open circles).

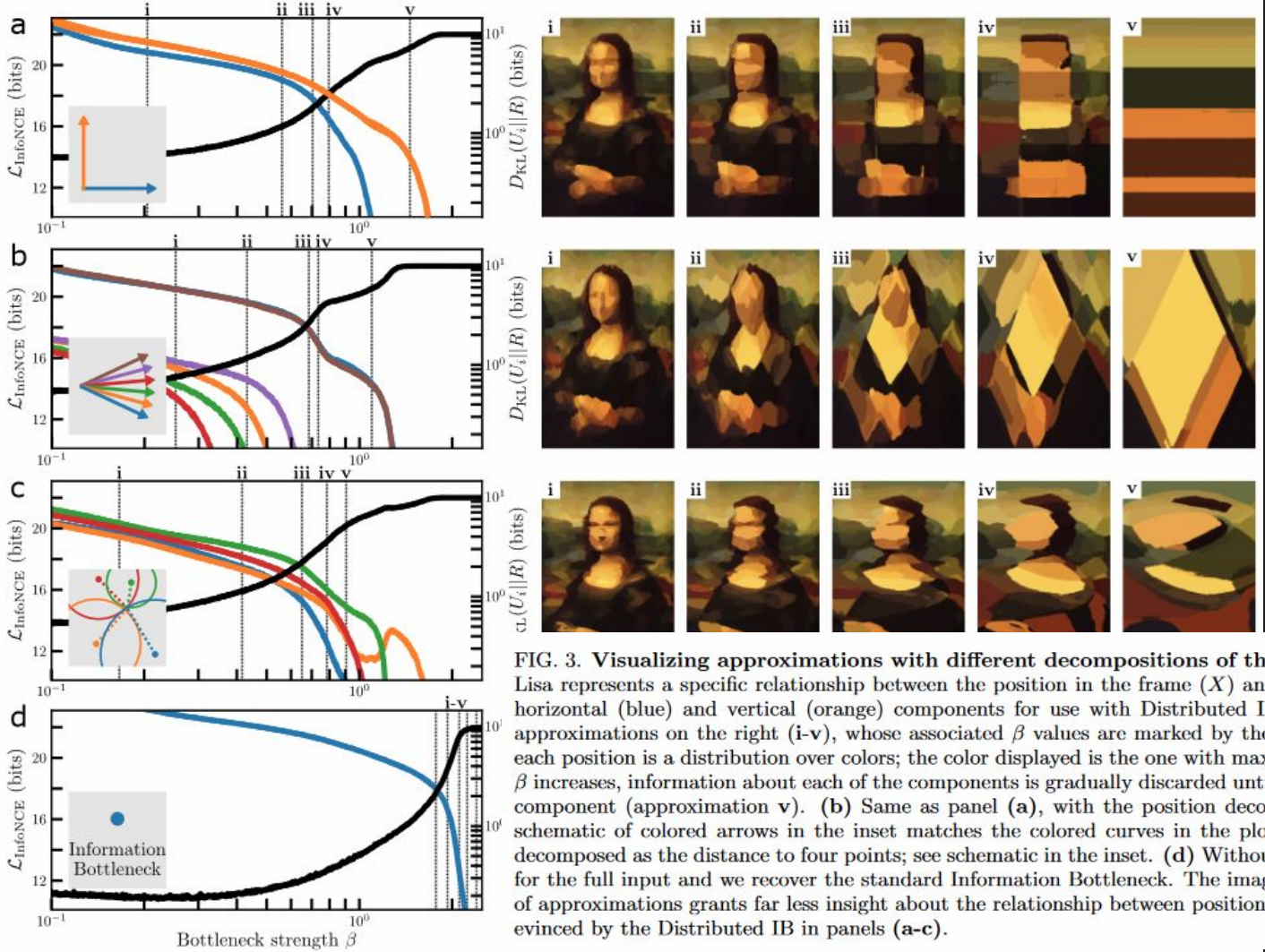


FIG. 3. **Visualizing approximations with different decompositions of the input.** **(a)** da Vinci's painting of the Mona Lisa represents a specific relationship between the position in the frame (X) and color (Y). The position is decomposed into horizontal (blue) and vertical (orange) components for use with Distributed IB (inset schematic). We display noteworthy approximations on the right **(i-v)**, whose associated β values are marked by the vertical bars in the plot. The prediction for each position is a distribution over colors; the color displayed is the one with maximum probability. As the bottleneck strength β increases, information about each of the components is gradually discarded until the only information comes from the vertical component (approximation **v**). **(b)** Same as panel **(a)**, with the position decomposed as the projection along six axes; the schematic of colored arrows in the inset matches the colored curves in the plot. **(c)** Same as panel **(a)**, with the position decomposed as the distance to four points; see schematic in the inset. **(d)** Without any decomposition there is only one channel for the full input and we recover the standard Information Bottleneck. The image degrades with increasing β, but the scheme of approximations grants far less insight about the relationship between position and color than the scheme of approximations evinced by the Distributed IB in panels **(a-c)**.

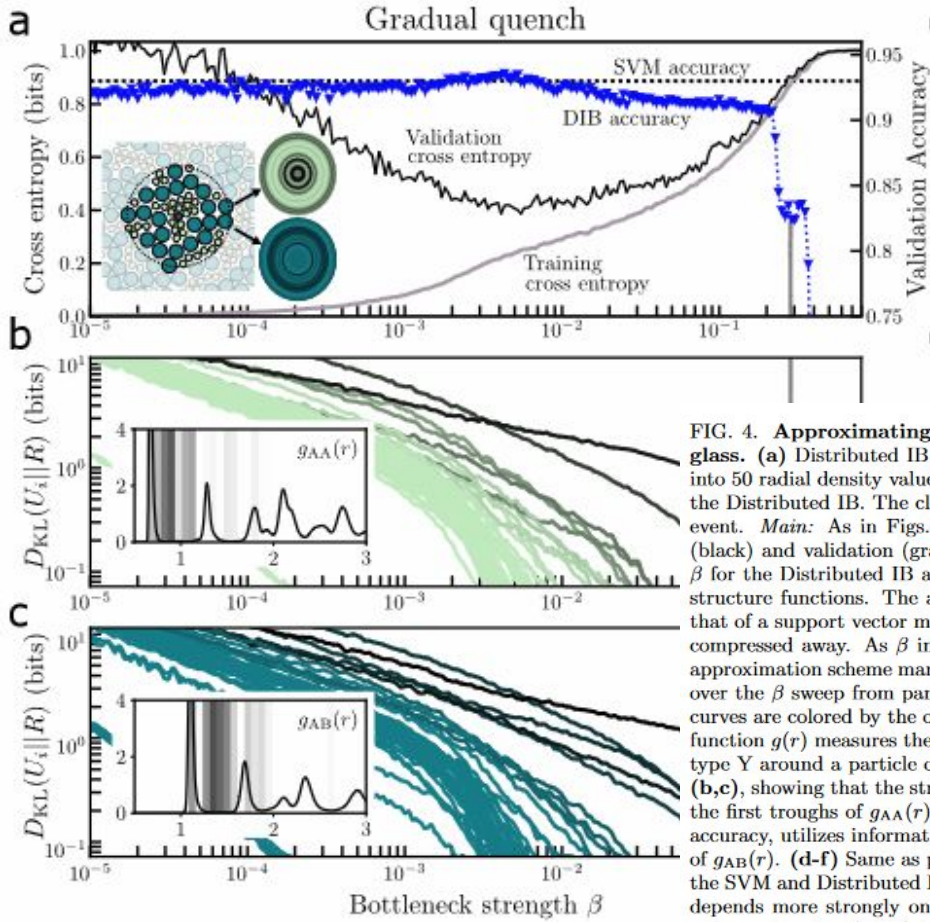
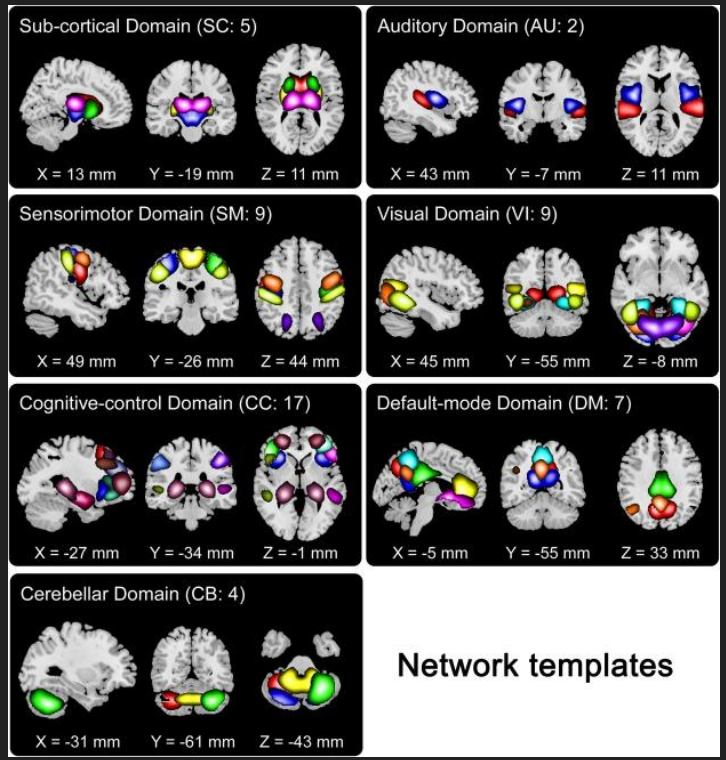
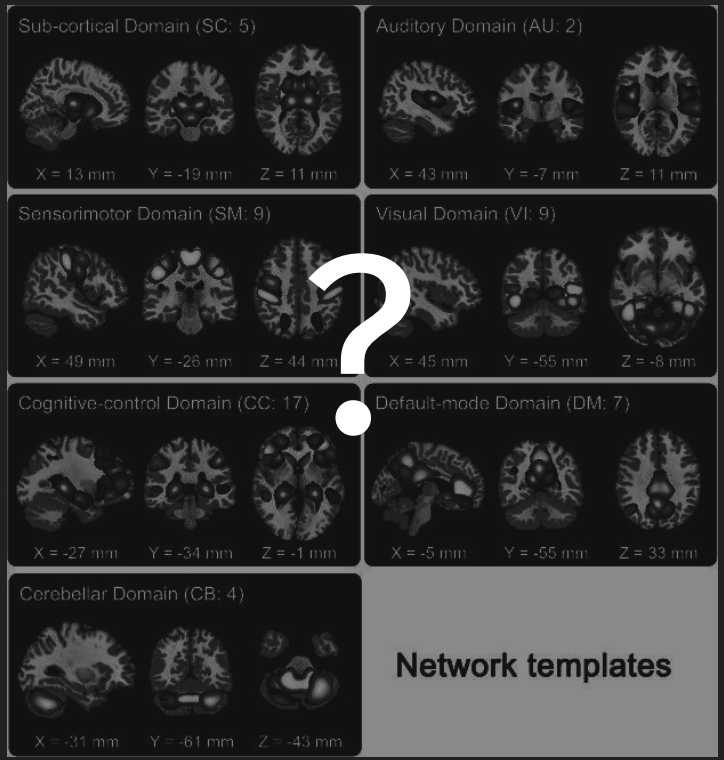


FIG. 4. **Approximating the relationship between static structure and imminent rearrangement in a sheared glass.** **(a)** Distributed IB on a gradually quenched glass. *Inset:* The local neighborhood around each particle is decomposed into 50 radial density values ("structure functions") for each of the large and small particle types, for use in both the SVM and the Distributed IB. The classification task is to predict whether the center particle is the locus of an imminent rearrangement event. *Main:* As in Figs. 2&3, the training error increases with the bottleneck strength β. While β is small, the training (black) and validation (gray) cross entropy errors (left vertical axis) display the hallmark behavior of overfitting. Increasing β for the Distributed IB acts as a regularizer and closes the gap between the two errors by restricting information about the structure functions. The accuracy on the validation set (right vertical axis) of the Distributed IB (blue) is comparable with that of a support vector machine trained on the same data (dotted black) until a majority of the structure functions have been compressed away. As β increases further, classification accuracy degrades until it drops precipitously, stopping briefly at an approximation scheme marked with the vertical gray bar. **(b)** The information maintained about each of the structure functions over the β sweep from panel **(a)** for the smaller type A particles. **(c)** Same as panel **(b)** for the larger type B particles. The curves are colored by the order with which they fall below a threshold value of the $D_{\mathrm{KL}}(U_i||R)$. *Insets:* The radial distribution function $g(r)$ measures the average density of radial shells for all particles in the system, with $g_{\mathrm{XY}}(r)$ the density of particles of type Y around a particle of type X at the origin. The radii are shaded according to the color scheme from the main panels of **(b,c)**, showing that the structure functions utilized for the coarser approximations by the Distributed IB are primarily those in the first troughs of $g_{\mathrm{AA}}(r)$ and $g_{\mathrm{AB}}(r)$. The coarsest approximation, highlighted in panel **(a)** and classifying with nearly 85% accuracy, utilizes information from only a single structure function involving the density of type B particles in the first trough of $g_{\mathrm{AB}}(r)$. **(d-f)** Same as panels **(a-c)**, with a glass prepared via rapid quench. Beyond poorer classification accuracy for both the SVM and Distributed IB, the Distributed IB sweep over β reveals how the connection between structure and rearrangement depends more strongly on information from many structure functions than in the gradual quench system. There is also no coarse approximation plateau as there was in panel **(a)**: with less information about the structure functions, all predictability of imminent rearrangement quickly degrades. Again, the insets of **(e-f)** show the most relevant structure functions lie in the troughs of $g_{\mathrm{AA}}(r)$ and $g_{\mathrm{AB}}(r)$ for the rapidly quenched glass.

Open Project: Application to ICA and Neuroimaging

Idea: we can find spatial decompositions using ICA, rather than arbitrary directional vectors. Or we can even do ICA-like analysis with DIB.





Questions and Further Discussion?