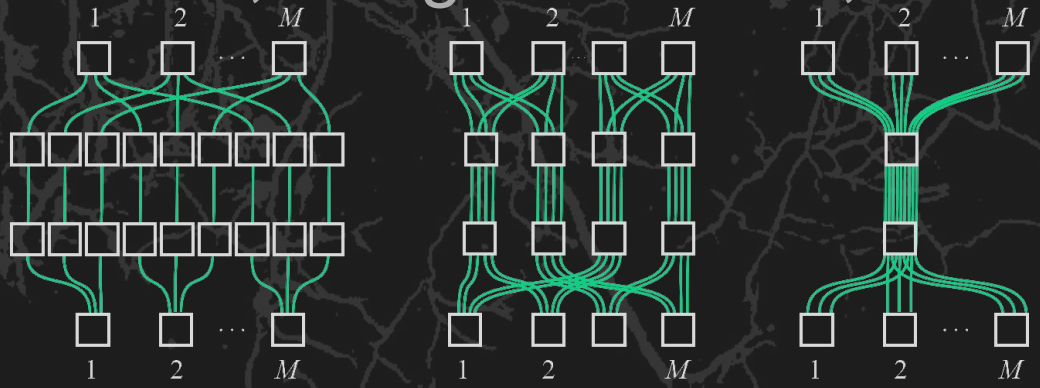
The Neural Race Reduction: Dynamics of Abstraction in Gated Networks

Andrew m. Saxe, Shagun Sodhani, Sam Lewallen



Do we understand Deep Neural Networks?

- We're beginning to understand optimization[1]
- Understanding of how the architecture (connectivity) affects behavior is limited[2]
- ImageNet Example - **wider networks** perform better on scene classes, and **deeper networks** are slightly more accurate are identifying consumer goods[3]

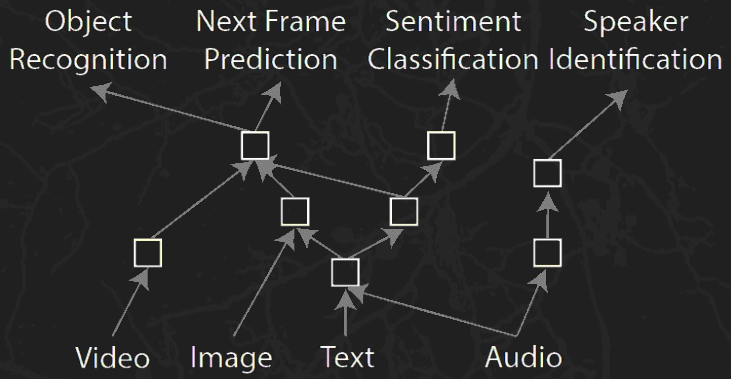
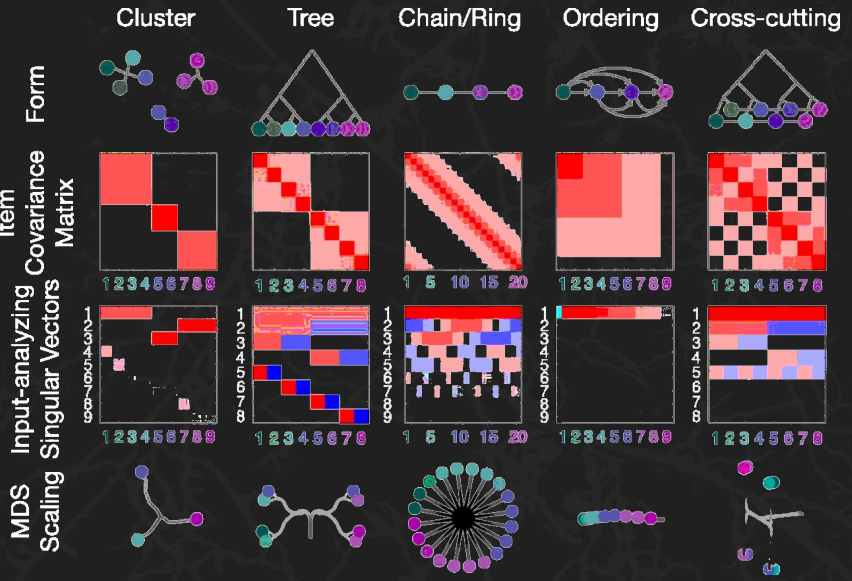


Figure 1. A multi-modal network composed of simple modules shared across modalities and tasks. How do shared modules and pathways impact representation learning and generalization?

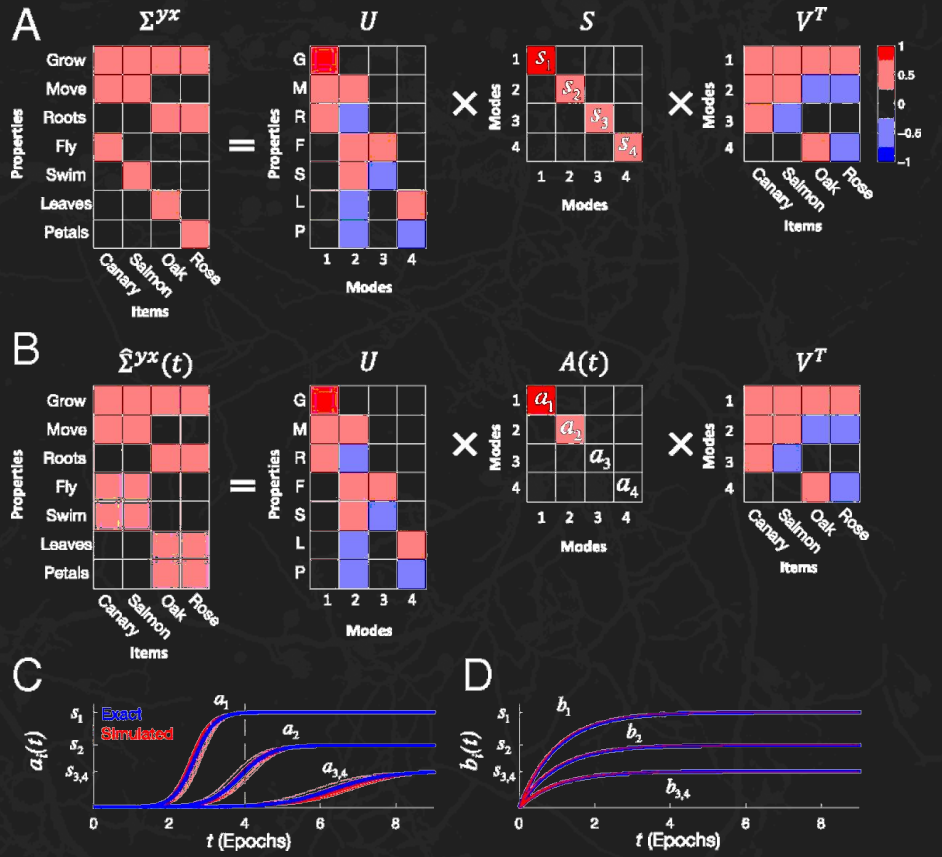
1) Patel et al., 2015; Carleo et al., 2019; Bahri et al., 2020; Arora et al., 2020; Roberts et al., 2021
2) Zagoruyko & Komodakis, 2016; Raghu et al., 2017; Chizat et al., 2019; Saxe et al., 2019; Tian et al., 2019
3) Nguyen et al. 2020

Background



Representation of explicit structural forms in a neural network. Each column shows a different structure. The first four columns correspond to pure structural forms, while the final column has cross-cutting structure. First row: The structure of the data generating PGM. Second row: The resulting item-covariance matrix arising from either data drawn from the PGM (first four columns) or designed by hand (final column). Third row: The input-analyzing singular vectors that will be learned by the linear neural network. Each vector is scaled by its singular value, showing its importance to representing the covariance matrix. Fourth row: MDS view of the development of internal representations.

4) Saxe et al. 2014, 2019



(A) SVD of input–output correlations. Associations between items and their properties are decomposed into modes. Each mode links a set of properties (a column of $\mathbf{U}$) with a set of items (a row of $\mathbf{V}^T$). The strength of the mode is encoded by the singular value (diagonal element of $\mathbf{S}$). (B) Network input–output map, analyzed via the SVD. The effective singular values (diagonal elements of $\mathbf{A}(t)$) evolve over time during learning. (C) Trajectories of the deep network's effective singular values $a_i(t)$. The black dashed line marks the point in time depicted in B. (D) Time-varying trajectories of a shallow network's effective singular values $b_i(t)$.

Gated Deep Linear Networks

$$h_v = g_v \sum_{q \in \mathbf{E}: t(q)=v} g_q W_q h_{s(q)}$$

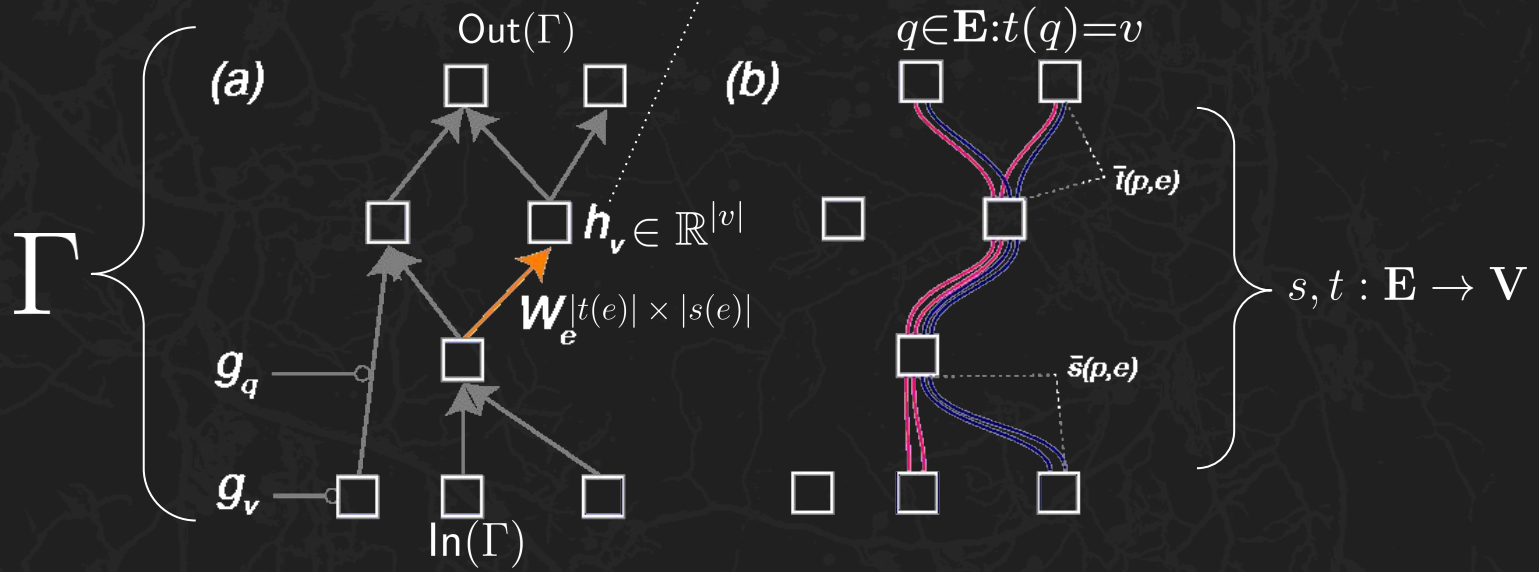


Figure 2. Formalism and notation. (a) The Gated Deep Linear Network applies gating variables to nodes and edges in an otherwise linear network. (b) The gradient for an edge (here using the orange edge in Panel (a) as an example) can be written in terms of paths through that edge (colored lines). Each path is broken into the component antecedent $\bar{s}(p, e)$ and subsequent $\bar{t}(p, e)$ to the edge.

Gradient Flow Dynamics

$$\tau \frac{d}{dt} W_e = -\frac{\partial \mathcal{L}(\{W\})}{\partial W_e} \quad \forall e \in E,$$
$$= \sum_{p \in \mathcal{P}(e)} W^T_{\bar{t}(p,e)} \mathcal{E}(p) W^T_{\bar{s}(p,e)}$$

$$\mathcal{E}(p) = \Sigma^{yx}(p) - \sum_{j \in \mathcal{T}(t(p))} W_j \Sigma^x(j,p)$$

$$\Sigma^{yx}(p) = \left\langle g_p y_{t(p)} x^T_{s(p)} \right\rangle_{y,x,g}$$
$$\Sigma^x(j,p) = \left\langle g_j x_{s(j)} x^T_{s(p)} g_p \right\rangle_{y,x,g}$$

Remember! s is antecedent, t is subsequent

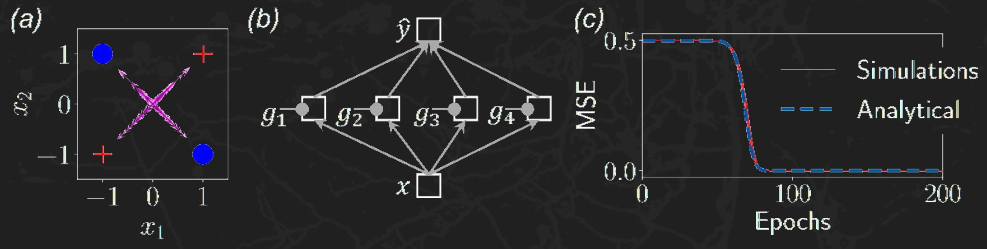


Figure 3. XoR solution dynamics. (a) The XoR task with positive (red) and negative (blue) examples. Input-to-hidden weights from ReLU simulations (magenta) reveal four functional cell types. (b) GDLN with four paths, each active on one example. (c) Simulations of ReLU dynamics from small weights (red, 10 repetitions) closely track analytical solutions in the GDLN. *Parameters:* $N_h = 128, \tau = 5/2, \sigma_0 = .0002$.

Gradient Flow Reduction

Weight matrix can have size N x M, but only min(M,N) singular values!

Change of variables - similar to Saxe et al. 2014

$$\Sigma^{yx}(p) = U_{t(p)} S(p) V^T_{s(p)}$$

$$\Sigma^{x}(j,p) = V_{s(j)} D(j,p) V^T_{s(p)}$$

$$W_e(t) = R_{t(e)} B_e(t) R^T_{s(e)} \quad \forall e$$

$$\tau \frac{d}{dt} B_e = \sum_{p \in \mathcal{P}(e)} B_{p \setminus e} \left[S(p) - \sum_{j \in \mathcal{T}(t(p))} B_j D(j,p) \right]$$

Example: Routing Setting

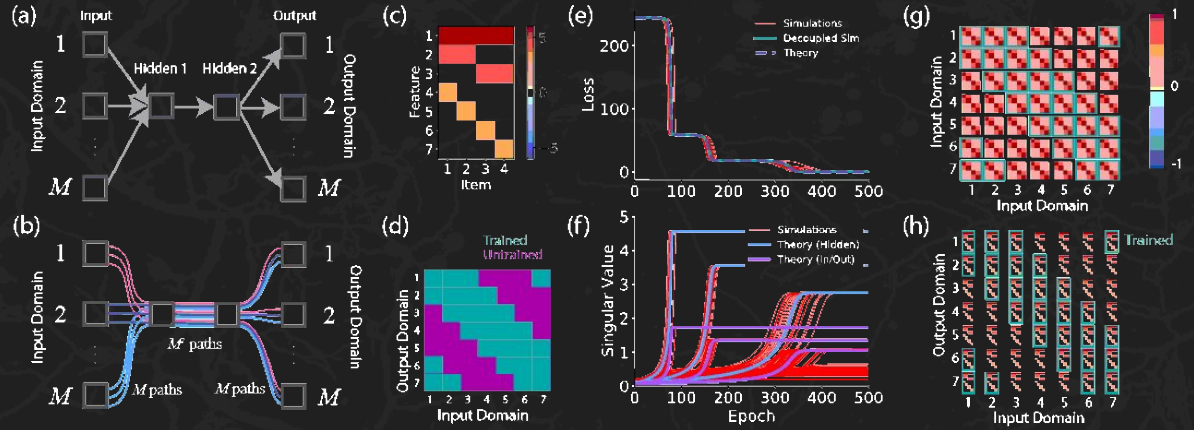


Figure 4. Pathway network solution dynamics. (a) The network contains M different input domains (each consisting of a bank of neurons), M different output domains, and two hidden layers. The task is to learn a mapping from each input domain to each output domain. The gating structure gates on one input and one output pathway. The hidden pathway is always on. (b) Gated network formalism. There are M^2 pathways through the network from input to output. All M^2 flow through the hidden weight matrix, while only M flow through each input or output weight matrix. This fact causes the hidden layer to learn faster. (c) Small example dataset with hierarchical structure. The task of the network is to produce the 7-dimensional output vector for each of four items. Inputs are random orthogonal vectors for each item. (d) Each input domain is trained with K output domains (here $K = 4$), such that some input-output routes are never seen in training. (e) Training loss dynamics for simulated networks from small random weights (red, 10 repetitions), simulated networks from decoupled initial conditions (green), and theoretical prediction from Eqn. 15 (blue). The theory matches the decoupled simulations exactly, and is a good approximation for small random weights. (f) The singular values of the hidden weight matrix (blue) are larger than those in input or output matrices by a factor $\sqrt{M}$. Theoretical predictions match simulations well, particularly for larger singular values. (g) Representational similarity (or kernel) matrix at the first hidden layer. Inputs from different domains are mapped to similar internal representations, revealing a shared representation even for input domains that are never trained with a common output. (h) Predicted output at the end of training. The network generalizes perfectly to input-output routes that were never seen during training. *Parameters:* $M = 7, K = 4, \lambda = .02, \sigma_0 = .2, N_h = 64$.

Neural Race Dynamics: Bias toward Shared Representations

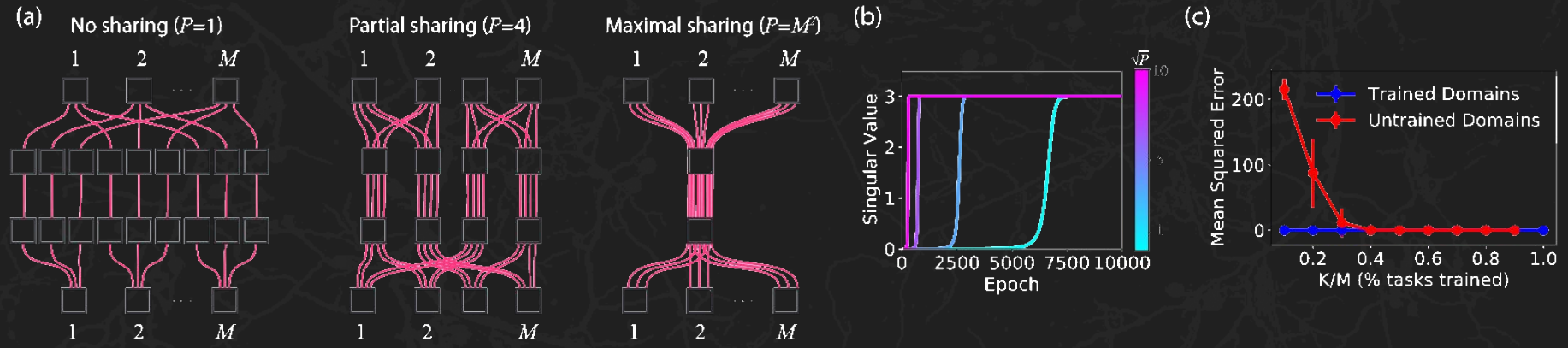


Figure 5. Pathway race dynamics. (a) The same routing task can be solved using a variety of gating schemes that differ in their use of shared representations. Every input-output combination can be given a dedicated pathway such that $P = 1$ tasks flow through each (left), groups of two input domains and two output domains can share a pathway such that $P = 4$ tasks flow through each (middle), or all $P = M^2$ tasks can run through a shared representation (right). (b) Singular value dynamics as a function of the number of pathways P flowing through the hidden layers. Networks that share more structure learn faster. Consequently, in a single network where subparts share structure to different degrees, the maximally shared dynamics dominate the race between pathways. (c) Error on trained (blue) and untrained (red) pathways as a function of the fraction of output domains K trained with each input domain M. When few outputs are trained per input domain, the race dynamics do not strongly favor shared structure and so error on untrained domains is large. When $\sim 40\%$ of output domains are trained with each input, the shared structure solutions are sufficiently faster to reliably dominate the race and yield generalization to untrained domains. *Parameters:* $M = 10, \lambda = .05/K, \sigma_0 = .2, N_h = 64$.

Impact of Initialization/ Zero-Shot Generalization

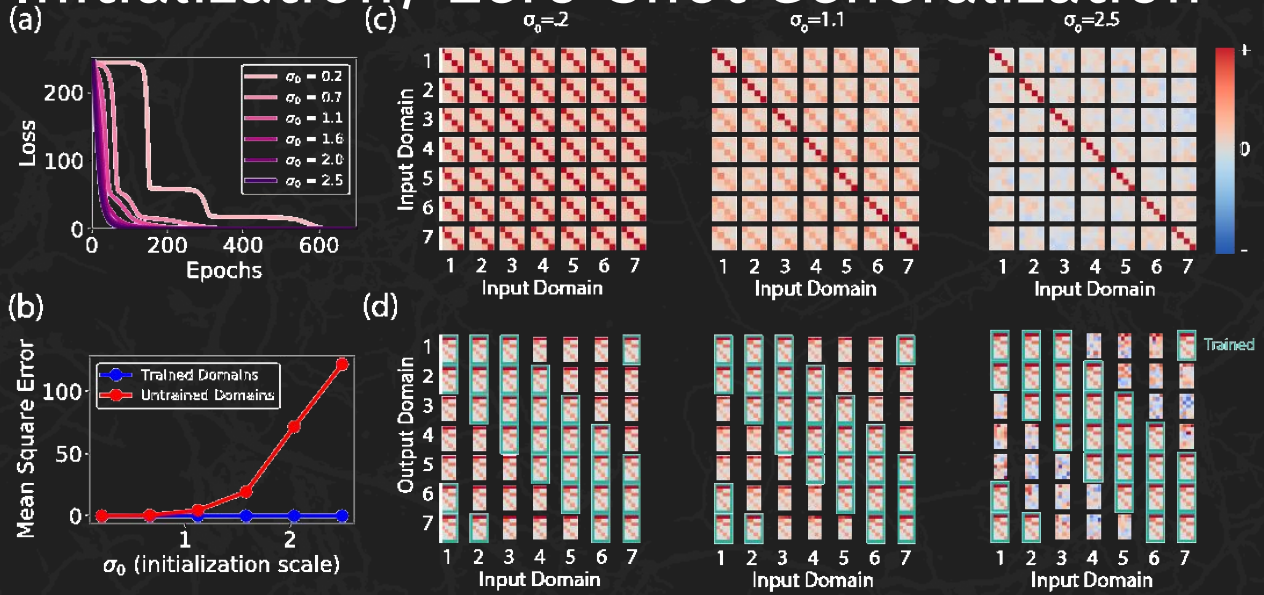


Figure 6. Neural race vs. NTK regime: Initialization, shared representation formation, and zero-shot generalization. (a) Training loss for pathway networks with different initialization scales σ_0. Small σ_0 yields pathway race dynamics with stage-like drops through training. Large σ_0 yields NTK-like dynamics with rapid, exponential learning curves. (b) Error for trained (blue) and untrained (red) input-output domain combinations as a function of initialization scale. While performance on trained domains is excellent for all scales, zero-shot generalization only emerges in the neural race regime. (c) Representational similarity between inputs presented to different domains, for small (left column), medium (middle column) and large (right column) initialization scales. At small initialization scales, internal representations in the first hidden layer are similar even across domains, indicating one common shared representation. Large initialization scales place the network in the NTK regime where random initial connectivity persists throughout learning, yielding distinct high-dimensional random representations for each domain. (d) Network output for all input-output combinations for three initialization scales (labeled in panel c). Because networks in the NTK regime do not learn a shared representation for different input domains, they do not generalize to untrained pathways.

Naturalistic Data sets

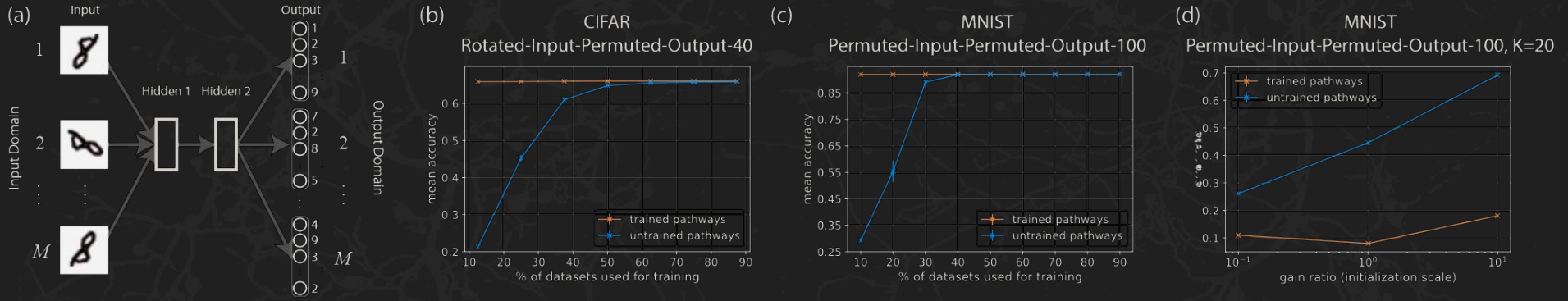
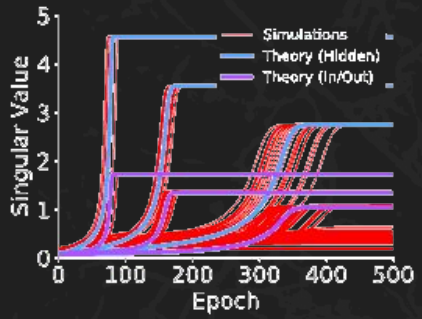
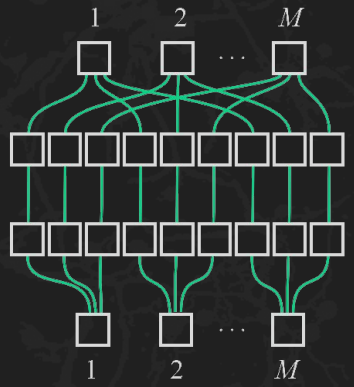
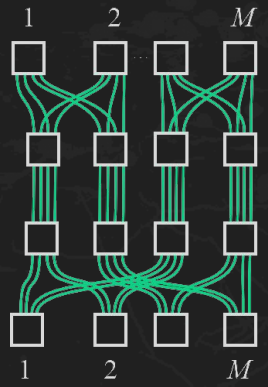


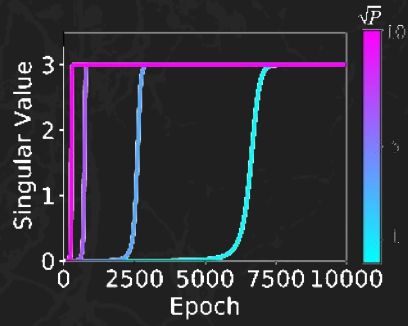
Figure 7. Experimental results. (a) Each input domain receives inputs that have been subjected to one of M input transformations. The target output for each output domain is also transformed by one of M output transformations. Here the visualization uses rotations in the input and permutations in the output. Only a subset of input-output transformation pairs are seen during training. (b) Error for trained (blue) and untrained (orange) input-output domain pairs as a function of the percentage of trained pathways (K/M) on the CIFAR dataset with $M^2 = 1600$ total tasks. (c) Error on MNIST with $M^2 = 10^4$ total tasks. Training accuracy is always high while zero-shot transfer to untrained pathways becomes as good as the training performance when $\approx$40% of pathways are trained. (d) Error as a function of initialization scale. While performance on trained domains is good for all scales, zero-shot generalization only emerges at small inits.

Discussion









Let's talk about the connections to

- Lottery Ticket Hypothesis[5]
- Low-Rank Dynamics
- Multi-Task Learning
- **Multi-Modal NeuroImaging**
  - I have a fun idea to use these to interpret modality interactions!
- Other Ideas...?

How can we improve/extend/criticize

- Parameter Tying
- Natural Results Non-Convincing?
- Non-ReLU-style Dynamics? Tanh?
- Hebbian Learning...?
- (Gated?) Spiking Network models...?
- Other issues...?

5) Frankle et al. 2018

Thanks!