A Linear Network Theory of Iterated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Language provides one of the primary examples of human's ability to systematically generalize — reasoning about new situations by combining aspects of previous experiences. Consequently modern machine learning has drawn much inspiration from linguistics. A recent example is iterated learning, a procedure where generations of networks learn from the output of earlier learners. The result is a refinement of the network's "language" or output labels for given inputs towards compositional structure. Yet, studies of iterated learning and its application to machine learning have remained empirical. Here we theoretically study the emergence of compositional language, and the ability of simple neural networks to leverage this compositionality to systematically generalize. We build on prior theoretical work on linear networks, which mathematically defines systematic generalization, by extending the analysis of shallow and deep linear network learning dynamics to the iterated learning procedure by deriving exact dynamics to the learning over generations. Our results confirm a long standing conjecture: that multiple generations of iterated learning are required for compositional structure to emerge, which can outperform a single generation network trained with optimal early-stopping. Finally, we show that IL requires depth in the network architecture to be effective and that IL is able to extract modules which systematically generalize.

1 Introduction

Deep learning techniques have made great strides in tasks like machine translation and language prediction, providing proof of principle that they can succeed in quasi-compositional domains. However, these methods are typically data hungry and the same networks often fail to generalize in even simple settings when training data are scarce (Lake & Baroni, 2018; Lake et al., 2019). *Systematically generalize*, leveraging specific learning experiences in diverse new settings (Lazaridou et al., 2018; Lake et al., 2019; Ren et al., 2019), has been proposed as a key feature of intelligent learning agents which can efficiently generalize to novel stimuli in their environment (Hockett & Hockett, 1960; Fodor & Pylyshyn, 1988; Hadley, 1993; Kirby et al., 2015; Lake et al., 2017). Empirically, the degree of systematicity in deep networks is influenced by many factors. One possibility is that the learning dynamics in a deep network could impart an implicit inductive bias toward compositional structure (Hupkes et al., 2020); however, a number of studies have identified situations where depth alone is insufficient for structured generalization (Lake & Baroni, 2018; Niklasson & Sharkey, 1992; Pollack, 1990; Phillips & Wiles, 1993; Jarvis et al., 2023). Another significant factor is architectural modularity, which can enable a system to generalize when modules are appropriately configured (Vani et al., 2021; Phillips, 1995; Andreas et al., 2016; Hu et al., 2017, 2018). However, identifying the right modularity through learning remains challenging (Bahdanau et al., 2019; Jarvis et al., 2023). A third possibility builds on iterated learning (IL), a method in which generations of agents train briefly on a language produced by their parent, and then generate a new language for their child (Kirby, 2001; Kalish et al., 2007; Kirby et al., 2015; Vani et al., 2021; Lu et al., 2020a,b). If compositional components are

Submitted to NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward. Do not distribute.

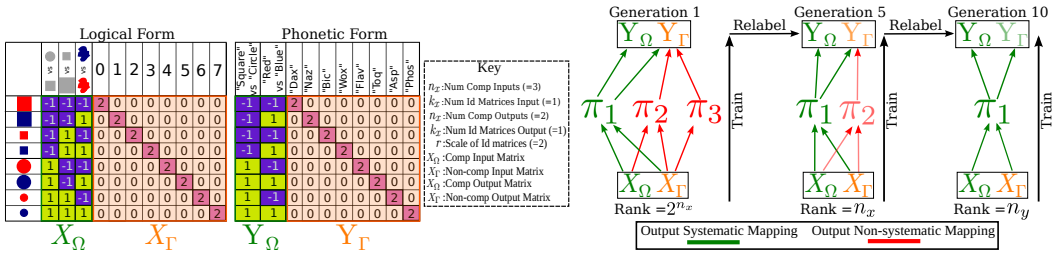

Figure 1: (Left) We schematize the setting with a space of datasets containing compositional (X_Ω) and non-compositional (X_Γ) features in the input (left panel). The task then is to map from these features to a phonetic form for each object which could be by the composition of descriptive words (Y_Ω) for example "the small red square" or by memorizing a name for each objects (Y_Γ) for example "the bic". (Right) The iterated learning procedure: Generations of agents learn from languages generated by their parent, and pass on their acquired language to their children. The portions of the language which are easier to learn are maintained over generations, while difficult language is lost. We demonstrates this process on a linear neural network and prove that IL is able to refine the language to depend on a minimal set of necessary singular values (π_1). This figure also summarizes the notation, structure and singular value decomposition of our space of datasets which is key for the theoretical results which follow.

easier to learn than non-compositional ones, this process can successively refine a language towards compositional structure, which has been hypothesized to be the cause of the compositional nature of natural language (Kirby, 2001; Kirby et al., 2008). In this work we aim to expand on the deep linear network framework of Saxe et al. (2019) (Saxe et al., 2019) and the formal analysis of systematicity of Jarvis et al. (2023) (Jarvis et al., 2023) to analyse the ability of IL to produce compositional language.

2 Iterated Learning Dynamics

The generalization abilities of deep networks depends on a complex interplay of learning dynamics (Saxe et al., 2014), architecture (Lake & Baroni, 2018), initialization (Geiger et al., 2020), and dataset structure (Jarvis et al., 2023). Prior work has demonstrated that gradient descent dynamics alone do not implicitly favour systematicity (Jarvis et al., 2023) for all but the most compositional of datasets. To establish whether *generations* of gradient descent learners have an implicit bias towards systematicity we obtain closed-form learning dynamics for neural networks in the IL procedure. We build on known exact solutions to the dynamics of learning from small random weights in deep linear networks (Saxe et al., 2014, 2019) to describe the full learning trajectory analytically.

In particular, consider a single hidden layer network computing output $\hat{y} = W^2W^1x$ in response to an input x, trained to minimize the quadratic loss $L(W^1, W^2) = \frac{1}{2}||Y - W^2W^1X||_2^2$ using full batch gradient descent. We review the derivation of the linear network dynamics in Appendix A. However, the main idea is to perform a change of variables so that we track the dynamics of the network's singular values rather than the individual weights. This does assume the network is feature learning such that its singular vectors align to those prescribed by the dataset statistics. This is a reasonable assumption from small initial weights (Saxe et al., 2019) and the singular vectors can be thought of conceptually as the features learned by the network. The trajectory of each network effective singular value ($\pi_\alpha(t)$) is described as

$$\pi_\alpha(t) = \frac{\lambda_\alpha/\delta_\alpha}{1 - \left(1 - \frac{\lambda_\alpha}{\delta_\alpha \pi_0}\right) \exp\left(\frac{-2\lambda_\alpha}{\tau} t\right)}. \tag{1}$$

These dynamics describe the singular value's trajectory which begins at the initial value π_0 when $t = 0$ and increases to $\lambda_\alpha/\delta_\alpha$ as $t \to \infty$. From these dynamics it is helpful to note that the time-course of the trajectory is only dependent on the input-output covariance matrix (Σ^{yx}) singular values (λ_α). Thus, the input covariance (Σ^x) (and its singular values δ_α), affects the stable point of the network singular values but not the rate of learning.

With IL each generation learns from the "language" acquired by the previous generation (Figure 1). To instantiate this setting, we start from a particular dataset, but halt training before full convergence after a pre-defined number of training steps. We then use the network's output (logits) as the target outputs for the next generation. From very early on in training, learning occurs along the modes

of variation determined by the dataset statistics. Consequently, the dataset's singular vectors, and the features learned by the network, will be maintained for all generations. It is merely the singular values or salience of the features which changes. Noting this fact permits straightforward analysis of iterated learning dynamics. Thus, for generation $G > 0$ of learning the asymptote of the network's mapping ($\lambda_\alpha^G/\delta_\alpha$) is equal to the effective singular value of the network at the end of the previous generation of training (π_α^{G-1}). Here λ_α^0 and δ_α^0 are the singular values from the original dataset. Thus, by a recursive application of Equation 1 we can model the full dynamics of iterated learning:

$$\pi_\alpha^G(t) = \frac{\lambda_\alpha^G/\delta_\alpha}{1 - \left(1 - \frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right) \exp\left(\frac{-2\lambda_\alpha^G}{\tau} t\right)} = \frac{\pi_\alpha^{G-1}}{1 - \left(1 - \frac{\pi_\alpha^{G-1}}{\pi_0}\right) \exp\left(\frac{-2\lambda_\alpha^G}{\tau} t\right)} \tag{2}$$

3 Theoretical Results

3.1 The Requirement of Multiple Generations

By ending a generation of training before the convergence of some effective singular values we will be decreasing the input-output singular values (λ_α) for the next generation. However, since these singular values also determine how quickly the mode is learned, this also means it will be learned slower for subsequent generations. The result is that it is eroded until it is removed from the "language" (has a singular value of 0). Since our analysis in this work relies on the relative speed at which modes start and finish learning it is helpful to introduce two new terminology. Firstly, by "escaping time" (denoted by $\hat{t}_\alpha$) we refer to the time taken for a mode (indexed by α) to grow meaningfully larger than 0: $\pi_\alpha > \rho$ for a small value of ρ. Secondly, "hitting time" (denoted by t_α^*) refers to the time taken for a mode (indexed by α) to converge to its final value: $\pi_\alpha - (\lambda_\alpha/\delta_\alpha) < \rho$. We derive explicit equations for $\hat{t}_\alpha$ and t_α^* in terms of the dataset singular values using Equation 2 in Appendix B. The equations themselves do not offer immediate insight beyond the concept they represent and so are omitted here. They are, however necessary for the proves of the theorem and observations which follow.

A lingering question in the use of IL has been whether multiple generations of learning is actually necessary. This is in contrast to a hypothetical optimal early stopping point which would provide all the same benefits as IL but within a single generation. It is important to note that IL must maintain all information which we do not wish to remove (maintained modes). There is naturally a trivial early-stopping time which removes all modes. To answer this question we present Theorem 3.1:

Theorem 3.1. *Given a dataset (X, Y), and assuming small random initial network parameters, a small learning rate ϵ and that removable modes have smaller singular values than maintained modes, $G > 0$ (having multiple generations of learners) is a **necessary** condition for guaranteed removal of only the desired modes of variation.*

Proof Sketch: To prove this result we are required to show that the escaping time for the removable modes is greater than the hitting time of the maintained modes. We show that this is not true in general for one generation (G=0) using a contradicting example. Secondly, we show that after some number of generations G the removable mode escaping time is guaranteed to be larger than the maintained mode's hitting time. The key step towards this is showing that the hitting time of the removable mode is larger than the hitting time of the maintained mode - a significantly easier condition than comparing hitting time and escaping time but which is enough for IL to be applicable. Once IL is applicable then as $G \to \infty$ the removable mode escaping time will become larger than the maintained mode hitting time, proving the theorem. The full proof is shown in Section B.

By assuming that the modes we aim to maintain are learned faster than removable modes we have imparted a preference in which modes of variation are learned and their relative ordering. This may appear to be a strong assumption, however, to maintain the slower learning modes would be a fundamental disagreement with IL as an algorithm. IL assumes that the fastest learning modes are systematic and provide the best generalization. Thus, for Theorem 3.1 our assumption of the ordering of the SVs is no more strict than assuming that IL is a valid algorithm for the dataset. We analyse the validity and limitations of the IL assumption that the quicker modes are systematic in Section 3.3.

3.2 The Requirement of Depth

A similar derivation of the IL learning dynamics can be done for a shallow network (no hidden layer). In this case the singular values of the model's mapping follow the trajectory:

$$\pi_\alpha^G(t) = \pi_\alpha^{G-1}\left(1 - \exp\left(-\delta_\alpha t/\tau\right)\right) + \pi_0 \exp\left(-\delta_\alpha t/\tau\right), \tag{3}$$

such that the time course depends on the singular values of the input covariance matrix, δ_α (Saxe et al., 2019). While deep networks show stage-like transitions which allow for one mode to be learned while another remains near 0; in shallow networks the modes show an exponential approach to their asymptote and all modes are learned at once. Thus, there will never be an opportunity for IL to remove a mode without also losing information on modes which we aim to maintain. See the simulated runs in Figure 2 for a visual depiction of this fact. This mean that for IL to be an effective procedure for the refinement of language, depth is required in the network architecture.

3.3 IL Uncovers Systematic Modules

To establish whether IL has a benefit for systematicity we must formalize a space of datasets which display the inductive biases of learning. We build on prior work which aimed to formalize systematicity (Jarvis et al., 2023) and provides such a space of datasets. To be applicable to the linguistic background of IL we phrase the space of datasets in terms of the mapping from logical forms (our internal, potentially semantic, representation of the world) to phonetic forms (words or sentences) commonly discussed in linguistics (Brighton & Kirby, 2006). The fundamental aspect of the analysis, however, remains the same: we use a space of datasets parametrized by the degree of compositional and non-compositional structure. We then use the closed-form SVD for all datasets in the space (written in terms of the dataset parameters) to establish how dataset structure affects the inductive bias of the neural network learning dynamics.

To formalize this setting Jarvis et al. (2023) define a parametric space of datasets with input and output matrices $X = [X_\Omega \; X_\Gamma]^T$ and $Y = [Y_\Omega \; Y_\Gamma]^T$ respectively, where $n_x, n_y, k_x, k_y, r \in \mathbb{Z}^+$ are the parameters that define a specific dataset. The *compositional input feature* matrix $X_\Omega \in \{-1, 1\}^{2^{n_x} \times n_x}$ consists of all binary patterns with n_x bits. Here n_x is a key parameter determining the number of bits in the compositional input structure. Overall, the dataset contains 2^{n_x} examples. The *non-compositional input feature* matrix $X_\Gamma = [rI_1 \; ... \; rI_{k_x}]$ consists of k_x scaled identity matrices, $I_i \in \{0, 1\}^{2^{n_x} \times 2^{n_x}}$ giving each datapoint a unique non-compositional identifying feature. Similar matrices are then defined for the output space. As described above, the network's total input-output mapping at all times in training is a function of the singular value decomposition of the dataset statistics. For all datasets in the space there are three distinct input-output covariance (Σ^{yx}) singular values λ_1, λ_2 and λ_3:

$$\lambda_1 = \left(\frac{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})}{2^{2n_x}} \right)^{\frac{1}{2}} \qquad \lambda_2 = \left(\frac{(k_x r^2 + 2^{n_x})(k_y r^2)}{2^{2n_x}} \right)^{\frac{1}{2}} \qquad \lambda_3 = \left(\frac{k_x k_y r^4}{2^{2n_x}} \right)^{\frac{1}{2}}$$

Note that the singular values are written in terms of the five dataset parameters, which allows for an analysis of how dataset structure influences the network training dynamics (Jarvis et al., 2023) and inductive bias of IL. Substituting these expressions into the dynamics equations we obtain equations for the networks mapping and full learning trajectories across all generations, at all times in training and for all datasets in the space. Appendix D depicts simulation to verify our theoretical results and we see exact agreement between the predicted and simulated dynamics. From this we can prove that for all datasets in the space the singular values will begin to learn in order. This means that IL can extract earlier singular values and remove later one. Since for the space of datasets the compositional input and output structure is only connected to the first mode π_1 (see Figure 1) IL is able to extract compositional, low-rank structure from the dataset. By the definition of systematicity in Jarvis et al. (2023), the reliance on low-rank structure here means that this module is generalizing systematically. This is a promising result and we hope that this will motivate more uses of IL in practical ML algorithms, for example like in the recent work of Ren et al. (2024).

Observation 3.2. *For all points in the space of datasets: $n_x, n_y, k_x, k_y, r \in \mathbb{Z}^+$ the input-output covariance matrix Σ^{yx} singular values will be ordered as: $\lambda_1 > \lambda_2 > \lambda_3$.*

Proof Sketch: The proof of this observation shows that there is no configuration of dataset parameters such that Equation λ_1 is not the largest value and λ_3 is not the smallest. We begin by assuming that this ordering holds and then simplify the expressions until we arrive at the requirement that $n_x, n_y, k_x, k_y, r \in \mathbb{Z}^+$ which is true by definition. The full proof is shown in Section C.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 39–48, 2016.

S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. *35th International Conference on Machine Learning, ICML 2018*, 1: 372–389, 2018. arXiv: 1802.06509 ISBN: 9781510867963.

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic generalization: What is required and can it be learned? In *International Conference on Learning Representations*, 2019.

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, January 1989. ISSN 08936080. doi: 10.1016/0893-6080(89)90014-2. URL `http://linkinghub.elsevier.com/retrieve/pii/0893608089900142`.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Henry Brighton and Simon Kirby. Understanding linguistic evolution by visualizing the emergence of topographic mappings. *Artificial life*, 12(2):229–242, 2006.

Jerry A Fodor and Zenon W Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, 1988.

K. Fukumizu. Effect of Batch Learning In Multilayer Neural Networks. In *Proceedings of the 5th International Conference on Neural Information Processing*, pp. 67–70, 1998.

Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020 (11):113301, 2020.

Robert F Hadley. Connectionism, explicit rules, and symbolic manipulation. *Minds and machines*, 3 (2):183–200, 1993.

Charles F Hockett and Charles D Hockett. The origin of speech. *Scientific American*, 203(3):88–97, 1960.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 804–813, 2017.

Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 53–69, 2018.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020.

Devon Jarvis, Richard Klein, Benjamin Rosman, and Andrew M Saxe. On the specialization of neural modules. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Fh97BDaR6I`.

Michael L Kalish, Thomas L Griffiths, and Stephan Lewandowsky. Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review*, 14(2):288–294, 2007.

Simon Kirby. Spontaneous evolution of linguistic structure-an iterated learning model of the emergence of regularity and irregularity. *IEEE Transactions on Evolutionary Computation*, 5(2): 102–110, 2001.

Simon Kirby, Hannah Cornish, and Kenny Smith. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686, 2008.

Simon Kirby, Monica Tamariz, Hannah Cornish, and Kenny Smith. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102, 2015.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 4487–4499. International Machine Learning Society (IMLS), 2018.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Brenden M Lake, Tal Linzen, and Marco Baroni. Human few-shot learning of compositional instructions. In *Proceedings of the 41st Annual Conference of the Cognitive Science Society*, 2019.

A.K. Lampinen and S. Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In T. Sainath (ed.), *International Conference on Learning Representations*, 2019. ISBN 0311-5518. doi: 10.1080/03115519808619195. URL `http://arxiv.org/abs/1809.10374`. arXiv: 1809.10374.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*, 2018.

Yuchen Lu, Soumye Singhal, Florian Strub, Aaron Courville, and Olivier Pietquin. Countering language drift with seeded iterated learning. In *International Conference on Machine Learning*, pp. 6437–6447. PMLR, 2020a.

Yuchen Lu, Soumye Singhal, Florian Strub, Olivier Pietquin, and Aaron Courville. Supervised seeded iterated learning for interactive language learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3962–3970, 2020b.

Lars Niklasson and Noel Sharkey. *Systematicity and generalisation in connectionist compositional representations*. Citeseer, 1992.

Steven Phillips and Janet Wiles. Exponential generalizations from a polynomial number of examples in a combinatorial domain. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 1, pp. 505–508. IEEE, 1993.

Steven Andrew Phillips. *Connectionism and the problem of systematicity*. PhD thesis, University of Queensland, 1995.

Jordan B Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–105, 1990.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2019.

Yi Ren, Samuel Lavoie, Michael Galkin, Danica J Sutherland, and Aaron C Courville. Improving compositional generalization using iterated learning and simplicial embeddings. *Advances in Neural Information Processing Systems*, 36, 2024.

A.M. Saxe, J.L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Y. Bengio and Y. LeCun (eds.), *International Conference on Learning Representations*, Banff, Canada, 2014. Oral presentation. arXiv: 1312.6120v3.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23): 11537–11546, 2019.

Ankit Vani, Max Schwarzer, Yuchen Lu, Eeshan Dhekane, and Aaron Courville. Iterated learning for emergent systematicity in vqa. In *International Conference on Learning Representations*, 2021.

A Linear Dynamics Review

While deep linear networks can only represent linear input-output mappings, the dynamics of learning change dramatically with the introduction of one or more hidden layers (Fukumizu, 1998; Saxe et al., 2014, 2019; Arora et al., 2018; Lampinen & Ganguli, 2019), and the learning problem becomes non-convex (Baldi & Hornik, 1989). They therefore serve as a tractable model of the influence of depth specifically on learning dynamics, which prior work has shown to impart a low-rank inductive bias on the linear mapping (Huh et al., 2021).

Consider a single hidden layer network computing output $\hat{y} = W^2W^1x$ in response to an input x, trained to minimize the quadratic loss $L(W^1, W^2) = \frac{1}{2}||Y - W^2W^1X||_2^2$ using full batch gradient descent. This gives the learning rules for each layer as $E[\Delta W^1] = \epsilon {W^2}^T(Y - W^2W^1X)X^T$ and $E[\Delta W^2] = \epsilon(Y - W^2W^1X)(W^1X)^T$. By using a small learning rate ϵ and taking the continuous time limit, the mean change in weights is given by $\tau \frac{d}{dt}W^1 = {W^2}^T(\Sigma^{yx} - W^2W^1\Sigma^x)$ and $\tau \frac{d}{dt}W^2 = (\Sigma^{yx} - W^2W^1\Sigma^x){W^1}^T$ where $\Sigma^x = E[XX^T]$ is the input correlation matrix, $\Sigma^{yx} = E[YX^T]$ is the input-output correlation matrix and $\tau = \frac{1}{P\epsilon}$ is the learning time constant for P inputs. Here, t measures units of learning epochs. It is helpful to note that since we are using a small learning rate the full batch gradient descent and stochastic gradient descent dynamics will be the same. Saxe et al. (2019) Saxe et al. (2019) has shown that the learning dynamics depend on the singular value decomposition of

$$\begin{aligned} \Sigma^x &= VDV^T = \sum_{\alpha=1}^{|X|} \delta_\alpha u^\alpha {v^\alpha}^T; \\ \Sigma^{yx} &= USV^T = \sum_{\alpha=1}^{\min(|X|,|Y|)} \lambda_\alpha u^\alpha {v^\alpha}^T \end{aligned} \tag{4}$$

where U and V are orthogonal matrices of singular vectors and S, D are diagonal matrices of singular values/eigenvalues. To solve for the dynamics we require that the right singular vectors V of Σ^{yx} are also the singular vectors of Σ^x. We also assume that $n_h > \min(|X|, |Y|)$ where n_h is the number of hidden neurons. If this is not the case then the model will only learn the top n_h singular values of the input-output mapping (Saxe et al., 2014). Given the SVDs of the two correlation matrices the learning dynamics can be described explicitly as

$$W^2(t)W^1(t) = UA(t)V^T = \sum_{\alpha=1}^{2^{n_x}} \pi_\alpha(t) u^\alpha v^{\alpha T} \tag{5}$$

where $A(t)$ is the effective singular value matrix of the network's mapping.

B Derivation of Escaping and Hitting Time

We will begin by using the definition of escaping time and substituting the mode dynamics (Equation 4) into this expression to obtain an expression for t:

$$
\begin{aligned}
\pi_\alpha^G &= \rho \\
\frac{\lambda_\alpha^G/\delta_\alpha}{1-\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right)} &= \rho \\
\frac{\lambda_\alpha^G/\delta_\alpha}{\rho} &= 1-\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\frac{\rho-\lambda_\alpha^G/\delta_\alpha}{\rho} &= \left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\frac{\rho-\lambda_\alpha^G/\delta_\alpha}{\rho\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)} &= \exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\log\left(\frac{\rho-\lambda_\alpha^G/\delta_\alpha}{\rho\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)}\right) &= \frac{-2\lambda_\alpha^G}{\tau}t \\
\frac{-\tau}{2\lambda_\alpha^G}\log\left(\frac{\rho-\lambda_\alpha^G/\delta_\alpha}{\rho\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)}\right) &= t \\
\frac{-\tau}{2\lambda_\alpha^G}\log\left(\frac{\rho\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \rho}\right)}{\rho\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}\right)}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{1-\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}}{1-\frac{\lambda_\alpha^G}{\delta_\alpha \rho}}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\frac{\lambda_\alpha^G}{\delta_\alpha \pi_0}-1}{\frac{\lambda_\alpha^G}{\delta_\alpha \rho}-1}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\frac{\lambda_\alpha^G-\delta_\alpha \pi_0}{\delta_\alpha \pi_0}}{\frac{\lambda_\alpha^G-\delta_\alpha \rho}{\delta_\alpha \rho}}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^G\delta_\alpha\rho-\delta_\alpha^2\pi_0\rho}{\lambda_\alpha^G\delta_\alpha\pi_0-\delta_\alpha^2\pi_0\rho}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^G\rho-\delta_\alpha\pi_0\rho}{\lambda_\alpha^G\pi_0-\delta_\alpha\pi_0\rho}\right) &= t
\end{aligned}
$$

The term inside of the log is always going to be greater than 1 as $\rho \geq \pi_0$. Thus the log is positive. In the extreme case where $\rho = \pi_0$ then the log evaluates to 0 as the internal fraction is 1, which makes sense as in this case the escaping time will be reached at initialization. We can perform a similar

computation for a modes hitting time.

$$
\begin{aligned}
(\lambda_\alpha^G/\delta_\alpha) - \pi_\alpha^G &= \rho \\
(\lambda_\alpha^G/\delta_\alpha) - \frac{\lambda_\alpha^G/\delta_\alpha}{1-\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right)} &= \rho \\
-\frac{\lambda_\alpha^G/\delta_\alpha}{1-\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right)} &= \rho - (\lambda_\alpha^G/\delta_\alpha) \\
-\frac{\lambda_\alpha^G/\delta_\alpha}{\rho-(\lambda_\alpha^G/\delta_\alpha)} &= 1-\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
-\frac{\lambda_\alpha^G/\delta_\alpha}{\rho-(\lambda_\alpha^G/\delta_\alpha)} - \frac{\rho-(\lambda_\alpha^G/\delta_\alpha)}{\rho-(\lambda_\alpha^G/\delta_\alpha)} &= -\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\frac{\rho}{\rho-(\lambda_\alpha^G/\delta_\alpha)} &= \left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)\exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\frac{\rho}{(\rho-(\lambda_\alpha^G/\delta_\alpha))\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)} &= \exp\left(\frac{-2\lambda_\alpha^G}{\tau}t\right) \\
\log\left(\frac{\rho}{(\rho-(\lambda_\alpha^G/\delta_\alpha))\left(1-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}\right)}\right) &= \frac{-2\lambda_\alpha^G}{\tau}t \\
\frac{-\tau}{2\lambda_\alpha^G}\log\left(\frac{\rho}{((\lambda_\alpha^G/\delta_\alpha)-\rho)\left(\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}-1\right)}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\left((\lambda_\alpha^G/\delta_\alpha)-\rho\right)\left(\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}-1\right)}{\rho}\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^{G^2}}{\delta_\alpha^2\pi_0\rho}-\frac{\lambda_\alpha^G}{\delta_\alpha\rho}-\frac{\lambda_\alpha^G}{\delta_\alpha\pi_0}+1\right) &= t \\
\frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^{G^2}-\delta_\alpha\lambda_\alpha^G\pi_0-\delta_\alpha\lambda_\alpha^G\rho+\delta_\alpha^2\pi_0\rho}{\delta_\alpha^2\pi_0\rho}\right) &= t
\end{aligned}
$$

Thus we can summarize the escaping time and hitting time for mode α as follows:

$$
\hat{t}_\alpha = \frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^G\rho-\delta_\alpha\pi_0\rho}{\lambda_\alpha^G\pi_0-\delta_\alpha\pi_0\rho}\right);\ t_\alpha^* = \frac{\tau}{2\lambda_\alpha^G}\log\left(\frac{\lambda_\alpha^{G^2}-\delta_\alpha\lambda_\alpha^G\pi_0-\delta_\alpha\lambda_\alpha^G\rho+\delta_\alpha^2\pi_0\rho}{\delta_\alpha^2\pi_0\rho}\right)
$$

Proof of Theorem 3.1

Theorem 3.1. Given a dataset (X, Y), and assuming small random initial network parameters, a small learning rate ϵ and that removable modes have smaller singular values than maintained modes, $G > 0$ (having multiple generations of learners) is a **necessary** condition for guaranteed removal of only the desired modes of variation.

We note that for now we make no assumptions on the dataset or singular values aside from their usual ordering. In subsequent sections we will make reference to the particular space of datasets. We now aim to show that there will always be a $G \geq 0$ for which the removable mode's hitting time will be larger than the maintained modes convergence time. To begin we need to show that the convergence time of the maintained mode is larger than the convergence time of the removable mode for all generations. This means that it is possible to find a time where the maintained mode has converged but the removable mode has not and IL is a valid algorithm for removing the removable mode. We may keep G general here as the generation does not change the proof since a removable

mode can only decrease in size. Thus, we begin by determining under what conditions the maintained mode will have an earlier convergence time than the removable modes:

$$t^*_{\alpha-1} > t^*_\alpha$$

$$\frac{\tau}{2\lambda^G_{\alpha-1}} \log\left(\frac{\lambda^{G^2}_{\alpha-1} - \delta_{\alpha-1}\lambda^G_{\alpha-1}\pi_0 - \delta_{\alpha-1}\lambda^G_{\alpha-1}\rho + \delta^2_{\alpha-1}\pi_0\rho}{\delta^2_{\alpha-1}\pi_0\rho}\right) > \frac{\tau}{2\lambda^G_\alpha} \log\left(\frac{\lambda^{G^2}_\alpha - \delta_\alpha\lambda^G_\alpha\pi_0 - \delta_\alpha\lambda^G_\alpha\rho + \delta^2_\alpha\pi_0\rho}{\delta^2_\alpha\pi_0\rho}\right)$$

We next substitute in the fact that $\pi_0 \to 0$ and $\rho \to 0$. However, we need to consider first that $\lambda^G_{\alpha-1}$ may be small (especially when IL is being applied). The above inequality will only hold when the log on the left of the inequality is positive. Thus, we first consider when the argument to the log is greater than 1:

$$\begin{aligned}
\frac{\lambda^{G^2}_{\alpha-1} - \delta_{\alpha-1}\lambda^G_{\alpha-1}\pi_0 - \delta_{\alpha-1}\lambda^G_{\alpha-1}\rho + \delta^2_{\alpha-1}\pi_0\rho}{\delta^2_{\alpha-1}\pi_0\rho} &> 1 \\
\frac{\lambda^{G^2}_{\alpha-1} - \delta_{\alpha-1}\lambda^G_{\alpha-1}\pi_0 - \delta_{\alpha-1}\lambda^G_{\alpha-1}\rho}{\delta^2_{\alpha-1}\pi_0\rho} + 1 &> 1 \\
\frac{\lambda^{G^2}_{\alpha-1} - \delta_{\alpha-1}\lambda^G_{\alpha-1}\pi_0 - \delta_{\alpha-1}\lambda^G_{\alpha-1}\rho}{\delta^2_{\alpha-1}\pi_0\rho} &> 0 \\
\lambda^G_{\alpha-1} - \delta_{\alpha-1}\pi_0 - \delta_{\alpha-1}\rho &> 0 \\
\frac{\lambda^G_{\alpha-1}}{\delta_{\alpha-1}} &> \pi_0 + \rho
\end{aligned}$$

Thus, for the log to be positive the final value of the mode must be larger than $\pi_0 + \rho$. This is a very easy constraint to meet and any mode with a final value less than $\pi_0 + \rho$ will have no real bearing on the output language. Additionally, in the limits of $\pi_0 \to 0$ and $\rho \to 0$ this just means that the final mode must be greater than 0 which is true by definition. We are free to substitute in $\pi_0 \to 0$ and $\rho \to 0$ to the original expression and it simplifies to the following where $c = \frac{1}{\pi_0\rho}$.

$$\begin{aligned}
\frac{1}{2\lambda^G_{\alpha-1}} \log\left(c\frac{\lambda^{G^2}_{\alpha-1}}{\delta^2_{\alpha-1}}\right) &> \frac{1}{2\lambda^G_\alpha} \log\left(c\frac{\lambda^{G^2}_\alpha}{\delta^2_\alpha}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\sqrt{c}\frac{\lambda^G_{\alpha-1}}{\delta_{\alpha-1}}\right) &> \frac{1}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right)
\end{aligned}$$

What this demonstrates is that, with small initial mode values and a high precision, the convergence time is inversely related to the size of the input-output covariance of the mode and proportional to the log of the final value of the mode itself. In essence, how long it takes learning to converge depends on how quickly learning happens and how much there is to learn. Noting that we can set c inside of the log and know that it is a very large value ($c \to \infty$) we can put the expression in a regime where the log derivative is near 0 and we can treat the two log expressions as constant and equal. We note that this is only valid where the log is positive which we have demonstrated is the case. How large we need to set c in practice depends entirely on the relative scale of $\lambda^G_{\alpha-1}/\delta_{\alpha-1}$ and $\lambda^G_\alpha/\delta_\alpha$. If $\lambda^G_{\alpha-1}/\delta_{\alpha-1}$ and $\lambda^G_\alpha/\delta_\alpha$ are of roughly the same magnitude then it may not even be necessary to set a large c for the inequality to hold, but this then depends on the relative scale of $\lambda^G_{\alpha-1}$ and λ^G_α in isolation too. What we are accounting for here is the case where a slower mode also has a much smaller final value, in which case it may still converge as quickly as the faster learning mode. The way to ensure these modes maintain their ordering is to set a smaller initial parameters and a smaller precision. This works because the dynamics have the sigmoidal training curve where overcoming the initial saddle point takes long and then there is a stage-like transition once the mode begins being learned. This simplifies the expression further to:

$$\frac{1}{\lambda^G_{\alpha-1}} \gtrapprox \frac{1}{\lambda^G_\alpha}$$

This is true by definition and so it is appropriate to apply IL for all generations. We will now consider the relationship between the removable modes hitting time and the maintained modes convergence

time.

$$\hat{t}_{\alpha-1} > t^*_\alpha$$
$$\frac{\tau}{2\lambda^G_{\alpha-1}} \log\left(\frac{\lambda^G_{\alpha-1}\rho - \delta_{\alpha-1}\pi_0\rho}{\lambda^G_{\alpha-1}\pi_0 - \delta_{\alpha-1}\pi_0\rho}\right) > \frac{\tau}{2\lambda^G_\alpha} \log\left(\frac{\lambda^{G^2}_\alpha - \delta_\alpha\lambda^G_\alpha\pi_0 - \delta_\alpha\lambda^G_\alpha\rho + \delta^2_\alpha\pi_0\rho}{\delta^2_\alpha\pi_0\rho}\right)$$

Once again, applying the usual limits of $\pi_0 \to 0$ and $\rho \to 0$ simplifies the expression to:

$$\begin{aligned}
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\lambda^G_{\alpha-1}\rho}{\lambda^G_{\alpha-1}\pi_0}\right) &> \frac{1}{\lambda^G_\alpha} \log\left(c\frac{\lambda^{G^2}_\alpha}{\delta^2_\alpha}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\rho}{\pi_0}\right) &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\rho}{\pi_0}\right) &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\rho^2 c\right) &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\frac{2}{\lambda^G_{\alpha-1}} \log\left(\rho\sqrt{c}\right) &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\rho\sqrt{c}\right) &> \frac{1}{\lambda^G_\alpha} \log\left(\frac{\lambda^G_\alpha}{\delta_\alpha}\sqrt{c}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\rho}{\sqrt{\rho\pi_0}}\right) &> \frac{1}{\lambda^G_\alpha} \log\left(\frac{\lambda^G_\alpha}{\delta_\alpha}\frac{1}{\sqrt{\rho\pi_0}}\right) \\
\frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\sqrt{\rho}}{\sqrt{\pi_0}}\right) &> \frac{1}{\lambda^G_\alpha} \log\left(\frac{\lambda^G_\alpha}{\delta_\alpha}\frac{1}{\sqrt{\rho\pi_0}}\right)
\end{aligned}$$

This expression is not true in general. For example if we set: $\lambda^G_{\alpha-1} = 1$, $\lambda^G_\alpha = \sqrt{2}$ and $\delta_\alpha = 1$ then we obtain:

$$\begin{aligned}
\frac{1}{2} \log\left(\frac{\rho}{\pi_0}\right) &> \frac{1}{2\sqrt{2}} \log\left(\frac{2}{\rho\pi_0}\right) \\
\log\left(\frac{\rho}{\pi_0}\right) &> \frac{1}{\sqrt{2}} \log\left(\frac{2}{\rho\pi_0}\right)
\end{aligned}$$

However $\rho \to 0$ and the left side of the expression is tending towards 0 while the right side is tending towards ∞. Thus, this is a contradiction. However, if we use the fact that $\lambda^G_{\alpha-1} \to 0$ as would be the case when the IL algorithm is applied, then regardless of the other dataset statistics the expression simplifies to:

$$\begin{aligned}
\lim_{G\to\infty} \frac{1}{\lambda^G_{\alpha-1}} \log\left(\frac{\rho}{\pi_0}\right) &> \lim_{G\to\infty} \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\frac{1}{\lambda^\infty_{\alpha-1}} \log\left(\frac{\rho}{\pi_0}\right) &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right) \\
\infty &> \frac{2}{\lambda^G_\alpha} \log\left(\sqrt{c}\frac{\lambda^G_\alpha}{\delta_\alpha}\right)
\end{aligned}$$

This is true by definition. In practice all that is required is to find some G for which the expression holds by a sufficient decrease in $\lambda^G_{\alpha-1}$. Thus, we have shown that it is always possible to reach a point where a removable mode's hitting time is higher than the maintained modes convergence time. This is not guaranteed to be the case for the first generation and may require multiple generations, which have shown to always be a viable strategy. As a consequence it is always possible to remove a removable mode while maintaining the maintained mode.

C Proof of Observation 3.2

Observation 3.2. For all points in the space of datasets: $n_x, n_y, k_x, k_y, r \in \mathbb{Z}^+$ the input-output covariance matrix Σ^{yx} singular values will be ordered as: $\lambda_1 > \lambda_2 > \lambda_3$.

Firstly we prove that $\lambda_1 > \lambda_2$:

$$\begin{aligned}
\lambda_1 &> \lambda_2 \\
\left(\frac{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})}{2^{2n_x}}\right)^{\frac{1}{2}} &> \left(\frac{(k_x r^2 + 2^{n_x})(k_y r^2)}{2^{2n_x}}\right)^{\frac{1}{2}} \\
\left((k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})\right) &> \left((k_x r^2 + 2^{n_x})(k_y r^2)\right) \\
k_y r^2 + 2^{n_x} &> k_y r^2 \\
2^{n_x} &> 0
\end{aligned}$$

$2^{n_x} > 0$ is true by definition since $n_x \in \mathbb{Z}^+$ and, thus, $\lambda_1 > \lambda_2$ for all points in our space of datasets. Now we prove that $\lambda_2 > \lambda_3$:

$$\begin{aligned}
\lambda_2 &> \lambda_3 \\
\left(\frac{(k_x r^2 + 2^{n_x})(k_y r^2)}{2^{2n_x}}\right)^{\frac{1}{2}} &> \left(\frac{k_x k_y r^4}{2^{2n_x}}\right)^{\frac{1}{2}} \\
(k_x r^2 + 2^{n_x})(k_y r^2) &> k_x k_y r^4 \\
k_x k_y r^4 + 2^{n_x} k_y r^2 &> k_x k_y r^4 \\
2^{n_x} k_y r^2 &> 0
\end{aligned}$$

$2^{n_x} k_y r^2 > 0$ is true by definition since $n_x, k_y, r \in \mathbb{Z}^+$ and, thus, $\lambda_2 > \lambda_3$ for all points in our space of datasets. Thus, using the transitivity of inequality: $\lambda_1 > \lambda_2 > \lambda_3$ for all points in the space of datasets.

D Simulations of IL Dynamics

To empirically verify our theoretical results we simulate the full training dynamics for deep and shallow linear networks trained using gradient descent on an instantiation from the space of datasets with parameters $n_x = 3, n_y = 2, k_x = 3, k_y = 1, r = 2$ (shown in Figure 2). While training, we compute the singular values of the network after each epoch. These simulations of the training dynamics for each unique singular value are then compared to the predicted dynamics. We also compute the Frobenius norms of portions of the network. These norms are functions of the singular values and summarize how entire portions of the input space connect to portions of the output space. Here we track how compositional/non-compositional inputs affect compositional/non-compositional outputs. The equations of these form Frobenius norms are shown below and were also first introduced in Jarvis et al. (2023). We see close agreement between the predicted and simulated trajectories[1]. Note the requirement of depth and multiple generations of IL to effectively remove a mode of variation without also losing information on other modes. The difference between deep and shallow network training dynamics can be seen by comparing the shape of the learning trajectories between Figures 2(a) and 2(c). Note how π_3 is removed with the deep network (Figure 2(a)) while π_1 and π_2 remain unchanged. In contrast, all modes are decreased with the shallow network (Figure 2(c)) but none are removed.

$$X_\Omega Y_\Omega^G\text{-Norm} = \left(\frac{2^{2n_x} n_y \pi_1^2(t)}{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})}\right)^{\frac{1}{2}} \tag{6}$$

$$X_\Gamma Y_\Omega^G\text{-Norm} = \left(\frac{2^{n_x} n_y k_x r^2 \pi_1^2(t)}{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})}\right)^{\frac{1}{2}} \tag{7}$$

$$X_\Omega Y_\Gamma^G\text{-Norm} = \left(\frac{2^{n_x} k_y n_y r^2 \pi_1^2(t)}{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})} + \frac{2^{n_x}(n_x - n_y)}{k_x r^2 + 2^{n_x}} \pi_2^2(t)\right)^{\frac{1}{2}} \tag{8}$$

[1]All experiments are run using the Jax library (Bradbury et al., 2018).

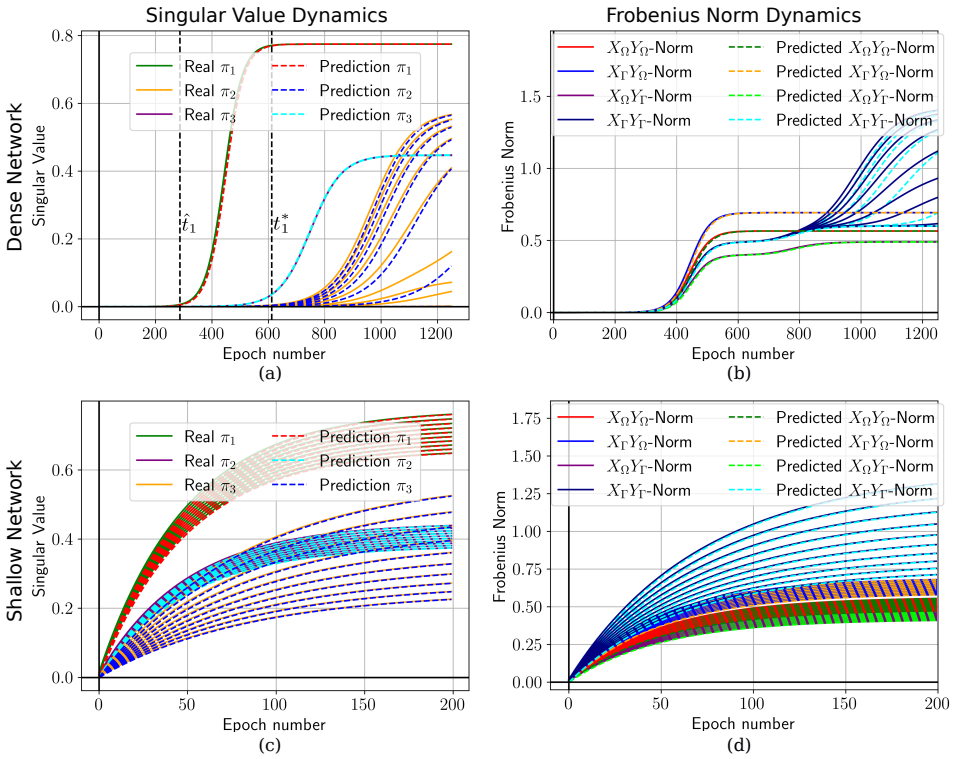

Figure 2: Analytical learning dynamics for deep (panels a-b) and shallow (panels c-d) linear networks. (a,c) Comparison of predicted and actual singular value trajectories over learning, for the three unique dataset singular values. $\hat{t}_1$ and t_1^* denote the escaping time and hitting time respectively for the first mode of variation with $\rho = 0.005$. (b,d) Comparison of predicted and actual Frobenius norms of the input-output mapping to/from compositional (X_Ω, Y_Ω) and non-compositional (X_Γ, Y_Γ) features. Deep networks show distinct stages of improvement over learning. However, at no point is a mapping learned which relies exclusively on compositional features or language. However, this setting depicts the progressive removal of the π_3 mode of variation over 10 generations. By the final generation of the dense network training the non-systematic norms exhibit two stage-like transitions corresponding to the learning of the two remaining modes of variation. The shallow network does not learn the modes in separate stages and so the removal of one distinct mode is impossible without simultaneously removing portions of all other modes. This demonstrates the theoretical observations from the dynamics of IL above. *Dataset Parameters*: $n_x = 3, n_y = 2, k_x = 3, k_y = 1, r = 2$.

$$X_\Gamma Y_\Gamma^G\text{-Norm} = \left(\frac{k_x k_y n_y r^4 \pi_1^2(t)}{(k_x r^2 + 2^{n_x})(k_y r^2 + 2^{n_x})} + \frac{(n_x - n_y) k_x r^2}{k_x r^2 + 2^{n_x}} \pi_2^2(t) + (2^{n_x} - n_x) \pi_3^2(t) \right)^{\frac{1}{2}} \tag{9}$$

E NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: There are three main contributions in the abstract and each is given its own section in our results. The introduction highlights the use of Saxe et al. (2019) and Jarvis et al. (2023) in particular for informing or methods and this is exactly what we use.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the approach are clearly stated where appropriate. We also review the linear dynamics and its assumptions in Section A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We state the assumptions and provide a proof sketch in the main text. Full proves are given in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the datasets, network architectures and training algorithms for all experiments in the work.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not. Finally, we release the code for all experiments (including bash scripts to make this easy) with a requirements file for reproducibility.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: This is a preprint, and so we omit code at this point. However we guide a reader to code release for Jarvis et al. (2023) which provides code to imitate the setup of this work and the linear network dynamics in that setting.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: These were described in the description of the experimental design and background on the GDLN paradigm as this requires full-batch gradient descent. All other details were mentioned or are relatively easy to determine, such as choosing learning rate. In this case the learning rate has quite a broad range of valid choices as long as it is not set too large.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: All experiments in this work have deterministic results. The random initialization of the network parameters does not affect this as they are set small enough to be very low variance.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The networks we use here are very small and train on the order of seconds on a single Nvidia 1080. The necessary compute is intuitive based off of the architecture descriptions.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This is a theory work and so we do not perceive of any potential ethics concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Naturally we believe that our work will be of benefit to the research community which could have positive societal impact. However, our work is theoretical and to comment on any positive or negative societal impacts here would be speculative and potentially irresponsible. Since there are no primary or immediate societal impacts we omit any further discuss.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: None of our datasets or models appears to have a potential for misuse requiring safeguards.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Citations are made to all works which directly influenced our own work or approaches. We also cite the Jax library as the framework which we used for the experiments. No other licensing issues arose.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new asset have been made beyond the academic contribution.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing or human subjects were used.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No such concerns arose requiring review or approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.