Early learning of the optimal constant solution in neural networks and humans

Anonymous authors
Paper under double-blind review

Abstract

Deep neural networks learn increasingly complex functions over the course of training. Here, we show both empirically and theoretically that learning of the target function is preceded by an early phase in which networks learn the optimal constant solution (OCS) – that is, initial model responses mirror the distribution of target labels, while entirely ignoring information provided in the input. Using a hierarchical category learning task, we derive exact solutions for learning dynamics in deep linear networks trained with bias terms. Even when initialized to zero, this simple architectural feature induces substantial changes in early dynamics. We identify hallmarks of this early OCS phase and illustrate how these signatures are observed in deep linear networks and larger, more complex (and nonlinear) convolutional neural networks solving a hierarchical learning task based on MNIST and CIFAR10. We explain these observations by proving that deep linear networks necessarily learn the OCS during early learning. To further probe the generality of our results, we train human learners over the course of three days on a structurally equivalent learning task. We then identify qualitative signatures of this early OCS phase in terms of true negative rates. Surprisingly, we find the same early reliance on the OCS in the behaviour of human learners. Finally, we show that learning of the OCS can emerge even in the absence of bias terms and is equivalently driven by generic correlations in the input data. Overall, our work suggests the OCS as a common learning principle in supervised, error-corrective learning, and suggests possible factors for its prevalence.

1 Introduction

Neural networks trained with stochastic gradient descent (SGD) exhibit various *simplicity biases*, where models tend to learn simple functions before more complex ones (Kalimeris et al., 2019; Rahaman et al., 2019). Simplicity biases hold significant theoretical interest as they provide an explanation for how deep networks generalize or fail to generalize in practice (Bhattamishra et al., 2023; Valle-Pérez et al., 2019; Zhang et al., 2021).

The characterisation of simplicity biases is still incomplete. Some explanations appeal to distributional properties of input data, pointing out that SGD progressively learns increasingly higher-order moments (Refinetti et al., 2023; Belrose et al., 2024). Other approaches focus directly on the evolution of the network function, proposing that networks initially learn a classifier highly correlated with a linear model. Importantly, networks continue to perform well on examples correctly classified by this simple function, even when overfitting in later training (Kalimeris et al., 2019). This implies that dynamical simplicity biases help models generalize, by locking in initial knowledge that is not erased or forgotten during later training (Braun et al., 2022; Kalimeris et al., 2019).

Deep linear networks have proven to be a valuable tool for studying simplicity biases. A key finding is that directions in the network function are learned in order of importance (Saxe et al., 2014; 2019). This phenomenon, known as *progressive differentiation*, connects modern deep learning theory to both to human child development and to the earliest connectionist models of semantic cognition (Rogers and McClelland, 2004; Rumelhart et al., 1986).

Our contribution proposes a connection between these works by characterizing networks in the *earliest* stages of learning in terms of input, output, and architecture. In the hierarchical setting by Saxe et al. (2019), we demonstrate both theoretically and empirically that neural networks initially

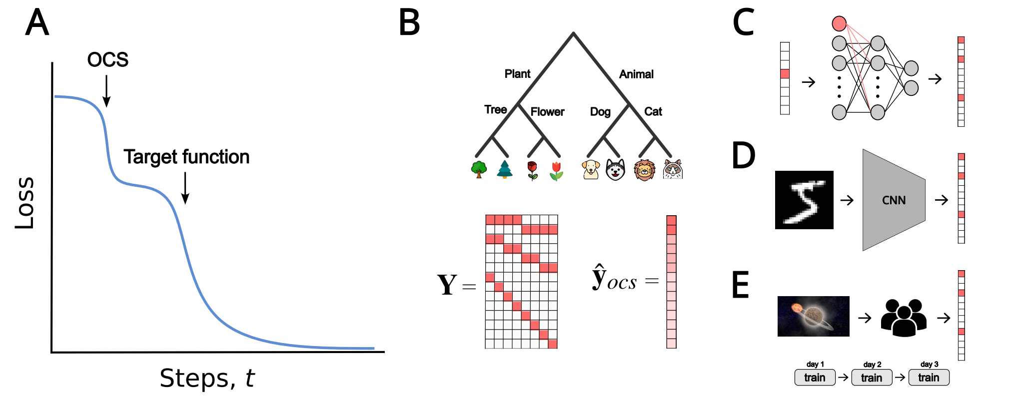


Figure 1: Early learning of the optimal constant solution (OCS). **A** Graphical illustration of our hypothesis, where learning of the target function is preceded by the early acquisition of the OCS. **B** *Top*: a graphical illustration of the hierarchical structure embedded in the outputs. *Bottom*: The full output data matrix $\mathbf{Y}$ used across different types of learners and the corresponding OCS solution $\hat{\mathbf{y}}_{ocs}$. **C** Illustration of experiments in linear networks with bias terms. **D** Illustration of our experiments with non-linear models and **E** Illustration of the task as adapted for humans.

learn via the output statistics of the data. This function has been termed the optimal constant solution (OCS) by Kang et al. (2024), who demonstrated that networks revert to the OCS when probed on out-of-distribution inputs. Here, we demonstrate and prove how linear networks, when equipped with these bias terms, necessarily learn the OCS early in training. Fig. 1A graphically illustrates this observation. We furthermore highlight the practical relevance of these results by examining early learning dynamics in complex, non-linear architectures.

Biological learners also display behaviours that imply the input-independent learning of output statistics. In probability matching, responses mirror the probabilities of rewarded actions (Herrnstein, 1961; Estes, 1964; Estes and Straughan, 1954). Learners often display non-stationary biases that are driven by the distribution of recent responses (Jones et al., 2015; Gold et al., 2008; Verplanck et al., 1952). In paired-associates learning accuracy can depend not only on a learned input-output mapping but also on knowledge of the task structure (Hawker, 1964; Bower, 1962). Humans also display simplicity biases and preferentially use simple over complex functions (Feldman, 2000; Goodman et al., 2008; Chater, 1996; Lombrozo, 2007; Feldman, 2003). However, relatively little attention has been devoted to the dynamics of these biases. We conduct experiments to determine whether humans replicate early reliance on the OCS.

1.1 Contributions

- We devise exact solutions for learning dynamics to analyse linear networks with bias in the input layer. Even when initialized at zero, this component substantially alters *early* learning dynamics.
- We empirically characterise early learning in these linear networks as being dominated by average output statistics. We explain this result with a theoretical analysis which reveals that average output statistics are always learned first when the network contains bias terms.
- We further highlight the practical relevance of these theoretical results in a hierarchical learning task for humans as well as complex, non-linear architectures by empirically demonstrating that all learners develop stereotypical response biases during early stages of training.
- On the basis of the developed theory we show that, in linear networks, early OCS learning can be induced by input correlations even in absence of bias terms. For natural datasets we empirically demonstrate that learning of the OCS can indeed be purely driven by generic correlations in the input data.

1.2 Related work

Deep linear networks. In deep linear networks analytical solutions have been obtained for certain initial conditions and datasets (Saxe et al., 2014; 2019; Braun et al., 2022; Fukumizu, 1998). Progress has also been made in understanding linear network loss landscapes (Baldi and Hornik, 1989) and generalisation ability (Lampinen and Ganguli, 2019). Despite their linearity these models display complex non-linear learning dynamics which reflect behaviours seen in non-linear models (Saxe et al., 2019). Moreover, learning dynamics in such simple models have been argued to qualitatively resemble phenomena observed in the cognitive development of humans (Saxe et al., 2019; Rogers and McClelland, 2004).

Biological response biases. Humans and animals routinely display response biases during perceptual learning and decision making tasks (Gold et al., 2008; Jones et al., 2015; Liebana Garcia et al., 2023; Amitay et al., 2014; Urai et al., 2019). In these tasks decisions are frequently made in sequences where responses and feedback steer decisions beyond the provided perceptual evidence (Jones et al., 2015; Fan et al., 2024; Gold et al., 2008; Verplanck et al., 1952; Sugrue et al., 2004). Non-stationary response biases can be driven by feedback on previous trials (Dutilh et al., 2012; Rabbitt and Rodgers, 1977) or might reflect global beliefs about the statistics of a task (Fan et al., 2024; Jones et al., 2015). Importantly, response biases are particularly pronounced in early learning (Jones et al., 2015; Gold et al., 2008; Liebana Garcia et al., 2023) and their influence appears to be strongest when uncertainty about the correct response is highest (Gold et al., 2008; Fan et al., 2024).

Simplicity biases in machine learning. Simplicity biases in neural networks have been studied extensively both theoretically (Bordelon et al., 2020; Mei et al., 2022) and empirically (Bhattamishra et al., 2023; Mingard et al., 2023). Work on the *distributional* simplicity bias emphasises the importance of input data and proposes that models learn via progressive exploitation of dataset moments (Refinetti et al., 2023; Belrose et al., 2024). On the other hand, neural networks have been found to express simpler functions during early training (Kalimeris et al., 2019; Refinetti et al., 2023; Belrose et al., 2024; Rahaman et al., 2019). Our work draws a connection between these findings and highlights how input statistics bias early learning towards output statistics.

1.3 Paper organisation

We initially review the linear network formalism in Section 2 on which we base our theoretical analysis. In Section 3 we derive learning dynamics for linear networks with bias terms trained on a classic hierarchical task and we document substantial changes in early dynamics. Section 4 characterizes this period of early learning empirically, and provides a theoretical explanation. We then in Section 4 validate the relevance of our findings for learning in complex models. Section 5 demonstrates the prevalence of early OCS learning in humans. Finally, Section 6 further probes generality by considering natural datasets and models that do not strictly fulfil the previous theoretical assumptions.

2 Linear network preliminaries

Here, we briefly review the analytical approach to learning dynamics in linear networks developed by Saxe et al. (2014; 2019). Consider a learning task in which a network is presented with input vectors $\mathbf{x}_i \in \mathbb{R}^{N_{in}}$ that are associated to output vectors $\mathbf{y}_i \in \mathbb{R}^{N_{out}}$. The total dataset consists of $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ with N samples. For our setting we consider two layer linear networks where the forward pass computes $\hat{\mathbf{y}}_i = \mathbf{W}^2\mathbf{W}^1\mathbf{x}_i$ and shallow networks with forward pass $\hat{\mathbf{y}}_i = \mathbf{W}^s\mathbf{x}_i$. Here weight matrices are of dimension $\mathbf{W}^1 \in \mathbb{R}^{N_{hid}\times N_{in}}$, $\mathbf{W}^2 \in \mathbb{R}^{N_{out}\times N_{hid}}$, and $\mathbf{W}^s \in \mathbb{R}^{N_{out}\times N_{in}}$. We train our networks to minimise a squared error loss of the form $\mathcal{L}(\hat{\mathbf{y}}) = \frac{1}{2}\sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|^2$.

We optimise networks using full batch-gradient descent in the gradient flow regime. When learning from small initial conditions, dynamics in these simple networks are solely dependent on the dataset input-output and input-input correlation matrices (Saxe et al., 2014). Using singular value decomposition (SVD), these matrices can be expressed as

$$\mathbf{\Sigma}^{yx} = \frac{1}{N}\mathbf{Y}\mathbf{X}^T = \mathbf{U}\mathbf{S}\mathbf{V}^T, \quad \mathbf{\Sigma}^{x} = \frac{1}{N}\mathbf{X}\mathbf{X}^T = \mathbf{V}\mathbf{D}\mathbf{V}^T. \tag{1}$$

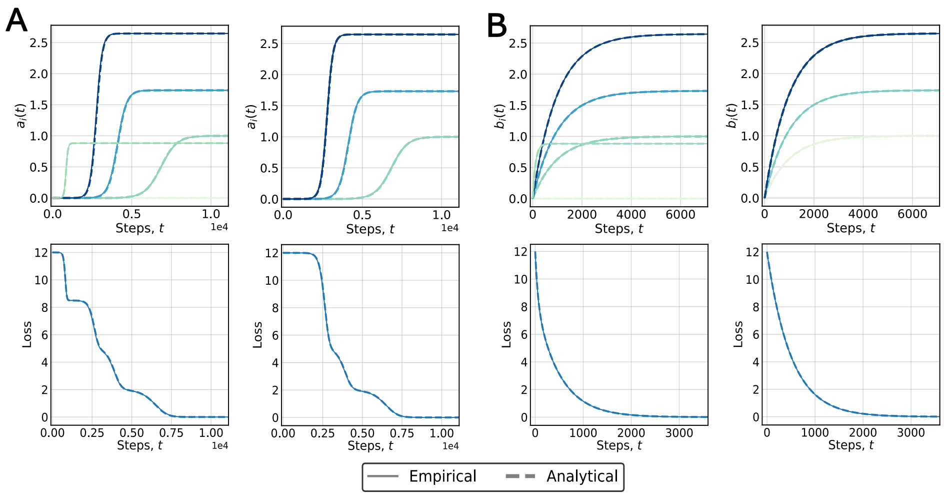


Figure 2: Exact learning dynamics. **A** Deep linear networks with bias (left) and without bias term (right). **B** Shallow linear networks with bias (left) and without bias term (right). *Top row:* Exact and simulated effective singular values $\mathbf{A}(t)$ and $\mathbf{B}(t)$ for deep and shallow linear networks respectively. Different $a_\alpha(t)$ and $b_\alpha(t)$ are color-coded according to their asymptote value with larger values as darker. *Bottom row:* Exact and simulated loss.

Here $\mathbf{X} \in \mathbb{R}^{N_{in} \times N}$ and $\mathbf{Y} \in \mathbb{R}^{N_{out} \times N}$contain the full set of input vectors and output vectors. Crucially, if the right singular vectors $\mathbf{V}^T$of $\mathbf{\Sigma}^{yx}$ diagonalise $\mathbf{\Sigma}^{x}$ (see Proposition 1) then the full evolution of network weights for deep and shallow networks through time can be described as

$$\mathbf{W}^2(t)\mathbf{W}^1(t) = \mathbf{U}\mathbf{A}(t)\mathbf{V}^T. \tag{2}$$

Here $\mathbf{A}(t)$ is a diagonal matrix. The evolution of these diagonal values $\mathbf{A}(t)_{\alpha\alpha} = a_\alpha(t)$ at each time-step t then follows a sigmoidal trajectory as expressed in Eq. (3). For shallow networks we can similarly describe the evolution of the weight matrix $\mathbf{W}^s(t)$ as $\mathbf{U}\mathbf{B}(t)\mathbf{V}^T$. Here the diagonal values $\mathbf{B}(t)_{\alpha\alpha} = b_\alpha(t)$ evolve as seen in Eq. (4)

$$a_\alpha(t) = \frac{s_\alpha / d_\alpha}{1 - (1 - \frac{s_\alpha}{d_\alpha a_0})e^{-\frac{2s_\alpha}{\tau}t}} \tag{3}$$

$$b_\alpha(t) = \frac{s_\alpha}{d_\alpha}(1 - e^{-\frac{d_\alpha}{\tau}t}) + b_0 e^{-\frac{d_\alpha}{\tau}t} \tag{4}$$

In Eq. (3) $s_\alpha = \mathbf{S}_{\alpha\alpha}$ and $d_\alpha = \mathbf{D}_{\alpha\alpha}$ denote the relevant singular values of $\mathbf{\Sigma}^{yx}$ and the eigenvalues of $\mathbf{\Sigma}^{x}$ respectively, a_0 are the singular values at initialisation, and $\tau = \frac{1}{N\epsilon}$ is the time constant where ϵ is the learning rate. In Eq. (4) b_0 is the initial condition given by the initialisation. Importantly, these relations reveal that singular values control learning speed. These solutions hinge on the diagonalisation of $\mathbf{\Sigma}^{x}$ through $\mathbf{V}$. Prior work has focused on the case of white inputs, i.e. $\mathbf{\Sigma}^{x} = \mathbf{I}_N$ where $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. The solution holds trivially as any $\mathbf{V}$ will orthogonalise $\mathbf{\Sigma}^{x}$ (Saxe et al., 2019). We discuss a relevant relaxation of this condition in Proposition 1. While solutions can be derived for some non-white inputs, little attention has been devoted to learning dynamics in these scenarios. We will show how these solutions apply when networks contain bias terms in the input layer.

3 Exact learning dynamics with bias terms

In this section, we derive exact learning dynamics in linear networks with bias terms and analyse the resulting changes in the dynamics. This extension to the theory by Saxe et al. (2014) forms the basis for our later discussion. For simplicity, we focus on input bias terms and uncorrelated data, but explore bias terms in other layers and correlated inputs in Appendix A.5.2 and Section 6, respectively.

Closed-form learning dynamics. We consider uncorrelated inputs $\mathbf{X} = \mathbf{I}_N$ where $\mathbf{I}_N$ denotes the $N \times N$ identity matrix. Our linear network with a bias term in the input layer will compute $\mathbf{W}^2(\tilde{\mathbf{W}}^1\mathbf{x}_i + \tilde{\mathbf{b}})$ where $\tilde{\mathbf{b}}$ are learnable bias terms.

A priori, it is unclear whether the diagonalisation of of $\mathbf{\Sigma}^x$ through $\mathbf{V}$ in Eq. (1) is possible in presence of bias terms. Here, we state the condition under which learning dynamics can be described in closed-form.

Proposition 1 (Feasibility of closed-form learning dynamics)**.** *For any input data $\mathbf{X} \in \mathbb{R}^{N_{in} \times N}$ and output data $\mathbf{Y} \in \mathbb{R}^{N_{out} \times N}$ it is possible to diagonalize $\mathbf{\Sigma}^x$ by the right singular vectors $\mathbf{V}$ of $\mathbf{\Sigma}^{yx}$ if $\mathbf{Y}^T\mathbf{Y}$ and $\mathbf{X}^T\mathbf{X}$ commute. The converse holds true only if $\mathbf{X}$ has a left inverse.*

A proof is given in Appendix A.5.4. We put this statement to use to assess the effect of a bias term on learning, building on the formalism from Section 2. To this end, we re-express the network weights as $\mathbf{W}^1 = \begin{bmatrix}\tilde{\mathbf{b}} & \tilde{\mathbf{W}}^1\end{bmatrix}$ with inputs defined as $\mathbf{x}_i = \begin{bmatrix}1 & \mathbf{I}_i^T\end{bmatrix}^T$ where $\mathbf{I}_i$ denotes the ith column of the $N \times N$ identity matrix (see Appendix A.5.1). To introduce a controlled setting in which to analyze the effect of bias terms, we now first consider a canonical hierarchical learning task while later sections of the paper will generalize our findings beyond this setting.

The hierarchical task. The hierarchical task requires learning a mapping from one-hot, input vectors to output vectors that are depicted in Fig. 1B. Hereby each output vector is "three-hot", i.e. the vector has three entries/labels. The hierarchical structure arises from the similarity between output vectors where some labels $y^m(\mathbf{x}_i)$ are more general and correspond to more than one input $\mathbf{x}_i$, while labels corresponding to the bottom of the hierarchy are specific to a single input vector $\mathbf{x}_i$. The task is motivated in the literature on semantic cognition and leverages the fact that semantic information is usually hierarchically structured (Rogers and McClelland, 2004). In Fig. 2 we depict exact learning trajectories for the hierarchically structured outputs from Fig. 1B.

Importantly, the introduction of a bias term $\mathbf{X} \to [\mathbf{1}_N\ \mathbf{X}]^T$ does not affect the commutativity of $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$ for the hierarchical dataset, as the constant mode $\mathbf{1}_N$ (i.e., a vector of 1s) is already an eigenvector to both these similarity matrices (see Appendix A.5.6). In consequence, the analytical solutions in Section 2 remain applicable. We generalize these considerations in Section 6 and Appendix A.5.6.

Fig. 2 shows that linear networks with bias terms have a distinctly different early learning phase when compared to vanilla linear networks. While both models converge to a zero loss solution, we observe that the final network function with bias terms contains an additional non-zero singular value with their associated singular vectors. We devote the next section to analyzing this change in the early dynamics.

4 Bias terms drive early learning towards the optimal constant solution

In this section, we qualitatively characterize what causes observed changes in early learning dynamics. We find that early learning dynamics are driven by average output statistics and provide a theoretical explanation. We then demonstrate the generality of this result by highlighting how early learning of average output statistics can be similarly observed in complex, non-linear architectures.

A naive strategy to learning is to minimise error over a set of samples while disregarding information conveyed by the input. Previous work has recently termed this network function the optimal constant solution (OCS) (Kang et al., 2024). The OCS can be formalised as $\hat{\mathbf{y}}_{ocs} = \operatorname{argmin}_{\hat{\mathbf{y}} \in \mathbb{R}^{N_{out}}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}_i)$ and represents the optimal function $\hat{\mathbf{y}}$ that is independent of input $\mathbf{x}_i$. For mean-squared error, it is straightforward to show that the minimiser is the average output $\hat{\mathbf{y}}_{ocs} = \frac{1}{N}\sum_i^N \mathbf{y}_i =: \bar{\mathbf{y}}$.

Setup. We train linear networks and Convolutional neural networks (CNN) on the hierarchical learning task illustrated in Fig. 1C and D respectively. For CNNs we design a "hierarchical MNIST" task whereby one-hot inputs are replaced with eight randomly sampled classes from MNIST (Li Deng, 2012). For the "hierarchical MNIST" task we used the started from ten digit classes provided by MNIST and then sampled 8 classes randomly. For each image in each class we then replaced

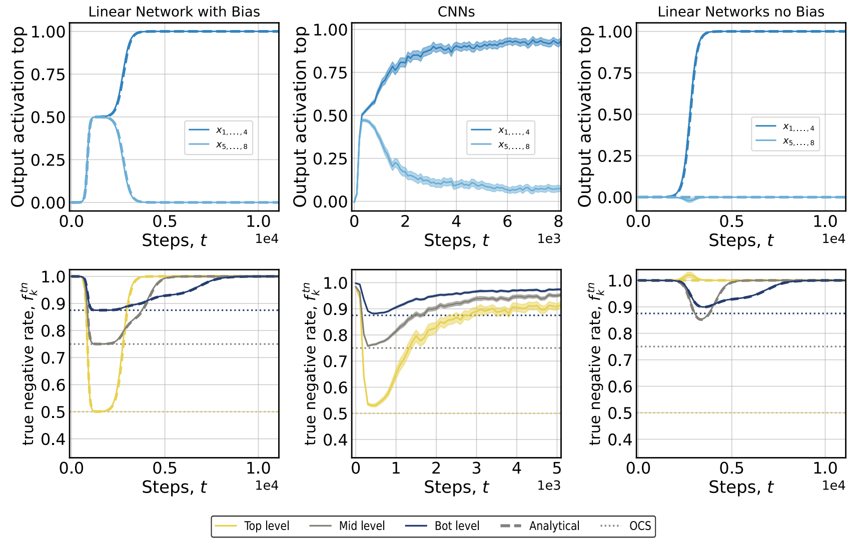


Figure 3: Early learning is driven to the OCS. *Top row:* Network predictions for a single output unit associated with the top level of of the hierarchy in response to all inputs $\mathbf{x}_i$ We see clearly how CNNs and linear networks *with bias* initially change responses while not differentiating between different inputs before learning the correct input output mapping. *Bottom row:* True negative rates, f_k^{tn} for the three hierarchical levels as indicated by colors. For CNNs and linear networks *with bias* Performance approaches levels expected under the OCS (dotted lines).

the default one-hot label corresponding to each class i with the corresponding hierarchical, "three-hot" label $\mathbf{y}_i$ seen in Fig. 1B. We use standard uniform Xavier initialization (Glorot and Bengio, 2010) and trained CNNs on an squared error loss. A full description of the CNN experiment and hyperparameter settings is deferred to Appendix A.8. We there also replicate our results with CIFAR-10 (Krizhevsky, 2009), non-hierarchical tasks, alternative loss functions, and CelebA (Liu et al., 2015) in Appendix A.9. We also show results for shallow networks in Appendix A.6.

To assess OCS learning we calculate true negative rates $f_k^{tn}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{(\mathbf{1}_N - \hat{\mathbf{y}}_k)^T(\mathbf{1}_N - \mathbf{y}_k)}{(\mathbf{1}_N - \mathbf{y}_k)^T(\mathbf{1}_N - \mathbf{y}_k)}$ for our task where the subscript k selects the vector slice corresponding to level k of the output hierarchy. We calculate the metric separately for the three hierarchical levels. Effectively, the metric describes how strongly model predictions $\hat{\mathbf{y}}$ align with the desired outputs $\mathbf{y}$ while focusing on zero entries only. The use of the metric is motivated by our desire to highlight how OCS learning is dependent on the distribution of labels in $\mathbf{Y}$ and effectively measures wrong beliefs about the presence of target labels across the different levels of the hierarchy. Furthermore, the metric enables later comparisons to human learners (further details in Appendix A.4).

4.1 Empirical Evidence

We identify three separate empirical observations that support early learning of the OCS:

Indifference. Linear networks and CNNs initially change outputs while not differentiating between input examples. In Fig. 3 (top) we show the empirical and analytical activation of an output unit associated with the highest level of the hierarchy for all $\mathbf{x}_i$. Networks with and without bias terms learn to differentiate inputs correctly. However, networks with bias terms produce input-independent, non-zero outputs in early training as would be expected under the OCS.

Performance. Networks with bias terms show an initial tendency to over-select labels associated with the top level of the hierarchy as seen in the true negative rate $f_k^{tn}(\mathbf{y}, \hat{\mathbf{y}})$ in Fig. 3 (bottom). Furthermore, linear networks and CNNs with bias terms almost exactly approach performance levels that would be provided by the OCS (dotted lines) for each of the three hierarchical levels. Linear network without bias terms do not produce this behaviour.

OCS alignment. The distance between outputs $\hat{\mathbf{y}}_i$ of linear and non-linear networks and $\mathbf{y}_{ocs}$ approaches zero in early training. Fig. 5 (top) shows how the L_1 distance of sample-averaged network outputs and the OCS approaches zero before later converging to the desired network function.

4.2 Theoretical explanation

In this section, we extend the linear network formalism to understand the mechanism behind early learning of the OCS. We first show how bias terms in the input layer are directly related to the OCS. Afterwards, we prove that the OCS is necessarily learned first in these settings.

The OCS is linked to shared properties. Having established the applicability of the linear network theory in Section 3 we now seek to understand how the early bias towards the OCS emerges. To this end, notice how bias terms can be written in terms of the constant eigenmode $\mathbf{1}_N$:

Proposition 2 (The OCS is linked to shared properties)**.** *If $\mathbf{1}_N$ is an eigenvector to the similarity matrix $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{N\times N}$, then the sample-average $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$ will be an eigenvector to the correlation matrix $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{N_{in}\times N_{in}}$ with identical eigenvalue λ. An analogous statement applies for $\mathbf{Y}^T\mathbf{Y}$ and $\mathbf{Y}\mathbf{Y}^T$. The converse does not hold true in general.*

We prove this statement in Appendix A.5.5. Importantly, it establishes a connection between the feature and sample dimensions of $\mathbf{X}$ and $\mathbf{Y}$. If $\mathbf{1}_N$ is an eigenvector to $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$ already, it implies that the addition of a bias term will directly add to its eigenvalue, $s_{ocs}^2 \to s_{ocs}^2 + 1$, even if it is initialized at zero. We show in Appendix A.5.6 that these assumptions on $\mathbf{X}$ and $\mathbf{Y}$ hold strictly for our hierarchical task design, and more generally relate to symmetry in the data (Appendix A.5.6). We discuss in Section 6 how this property extends to natural datasets where exact symmetry is absent.

Crucially, it now follows from Proposition 2 that the time-dependent network correlation $\hat{\mathbf{\Sigma}}^{yx}(t) = \mathbf{U}\mathbf{A}(t)\mathbf{V}^T$ in Eq. (2) will contain a strongly amplified OCS mode $a_{ocs}(t)\mathbf{u}_{ocs}\mathbf{v}_{ocs}^T = a_{ocs}(t)\bar{\mathbf{y}}\bar{\mathbf{x}}^T$ by virtue of the modified singular value $\sqrt{s_{ocs}^2+1}$ entering Eq. (2) and thereby the network function. Consequently, learning dynamics will be driven by the outer product of average input and output data. Moreover, this implies that given some input $\mathbf{x}_i$ to Eq. (2), the network's OCS mode contributes

$$\hat{\mathbf{y}}_{ocs}(\mathbf{x}_i) = a_{ocs}(t)\mathbf{u}_{ocs}\mathbf{v}_{ocs}^T\mathbf{x}_i = a_{ocs}(t)\bar{\mathbf{y}}\bar{\mathbf{x}}^T\mathbf{x}_i \propto \bar{\mathbf{y}}. \tag{5}$$

The OCS mode in the time-dependent network function will hence necessarily drive responses towards average output statistics. Note that Eq. (5) also highlights that the more an input example is aligned to average inputs, the more the network's responses will reflect average outputs. In particular, this makes the expected output $\mathbb{E}_{\mathbf{x}}[\hat{\mathbf{y}}(\mathbf{x})] \propto \bar{\mathbf{y}}$. Throughout learning, the evolution of $a_{ocs}(t)$ and scale-dependent alignment of $\mathbf{x}_i$ and $\bar{\mathbf{x}}$ will determine the network's reliance on the OCS mode.

Early learning is biased by the OCS mode. We established that network responses are driven by average output statistics $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, but why are *early* dynamics in particular influenced by the OCS? The learning speed of the SVD modes in the time-dependent network function are controlled by the magnitude of singular values s_α as seen in Eq. (3).

Theorem 1 (Early learning is biased by the OCS mode)**.** *If $\mathbf{1}_N$ is a joint non-degenerate eigenvector to positive input and output similarity matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$, the OCS mode $s_{ocs}\bar{\mathbf{y}}\bar{\mathbf{x}}^T$ will have leading spectral weight $s_0 \equiv s_{ocs}$ in the SVD of the input-output correlation matrix $\mathbf{\Sigma}^{yx}$.*

We prove this statement with help of the Perron-Frobenius theorem (Perron, 1907) in Appendix A.5.7. Consequently, the optimal constant mode is learned at a faster rate than remaining SVD components and transiently dominates the early network function. Notably, this applies to our task data $\mathbf{Y}^T\mathbf{Y}$ (see Appendix A.5.6) and leads to characteristic learning signatures observed in Fig. 3.
Theorem 1 hinges on the constant eigenvector $\mathbf{1}_N$ being present in the data. We later provide empirical (Fig. 6 and Appendix A.9) and theoretical (Appendix A.5.6) arguments that this assumption is approximately fulfilled in a variety of cases.

To recapitulate this section: We first rephrased a learnable bias term in the architecture as a shared feature in the input data. We then found that the associated singular value in Eq. (2) drives the learned network function towards the OCS (Eq. (5)). Finally, we proved that the bias affects *early* learning. In Appendix A.5.3, we summarize these results through the neural tangent kernel. Overall, these results demonstrate how architectural bias terms induce early OCS learning.

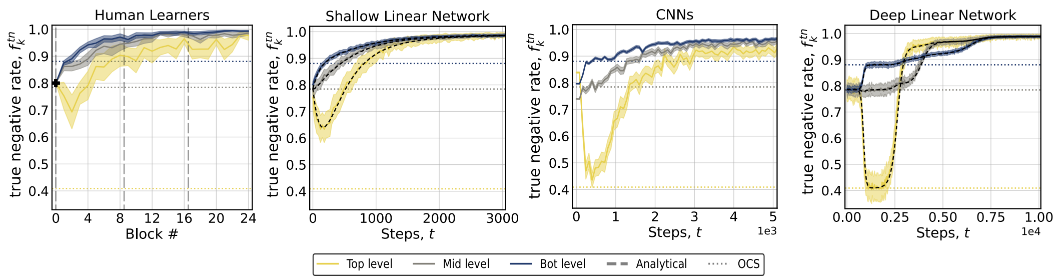


Figure 4: Early response bias towards the OCS across learners in the hierarchical learning task. True negative rates, f_k^{tn} for the three hierarchical levels as indicated by colors for biological and artificial learners (with bias terms). Dotted lines represent performances expected under the OCS. Observe that all learners show a transient bias towards the OCS. Dashed vertical gray lines indicates breaks between days for human learners.

5 Similarities between linear networks, humans, and complex models

In this section, we demonstrate how human learners, linear networks, and non-linear architectures show strong similarities in their early learning on the hierarchical task displayed in Fig. 1.

Setup. The hierarchical learning task has previously been used extensively in the study of semantic cognition (Rogers and McClelland, 2004) and requires learners to develop a hierarchical one-to-many mapping as seen in Fig. 1B. We adapted the task for human learners while maintaining the underlying structure: Input stimuli were represented as different classes of planets and output labels were represented as a set of plant images (see Fig. 1E and Fig. 7). We also trained CNNs as in Section 4. Importantly, the hierarchical structure results in a non-uniform distribution of labels with average labels equal to $\mathbf{y}_{ocs}$. Human learners received supervised training over three days. A full description of the experimental paradigm is given Appendix A.2. We then compute true negative rate $f_k^{tn}(\mathbf{y}, \hat{\mathbf{y}})$ as in Section 4 while splitting performance across the hierarchical levels as before.

Neural networks produced continuous outputs in $\mathbb{R}^{N_{out}}$ while humans responded via discrete button clicks in $\{0,1\}^{N_{out}}$. As we do not have access to "human logits" before response execution we discretized network responses to enable comparison. We treat network responses in $\hat{\mathbf{y}}$ as logits from which we then sampled responses in $\{0,1\}^{N_{out}}$. Full procedure details are given in Appendix A.3.

Results. The key results of our experiments are presented in Fig. 4. Intriguingly, we find that human learners, linear networks, and CNNs all display characteristic early response biases. Note that chance true negative rate is equal between all three levels of the hierarchy. Biological as well as artificial learners display an initial "drop" in true negative rate at the top level of the output hierarchy. The result indicates a general lack of specificity and an overly liberal response criterion for output labels on the top level of the hierarchy. To appreciate the significance of this result it is important to understand that the task can be learned without the development of these early response biases: In particular, linear networks without bias terms do not show this behaviour (see Appendix A.7). Surprisingly, the human response signature demonstrates that these learners, just as artifical networks, display an early bias towards the OCS. We conjecture that early learning of the OCS might be a general phenomenon that emerges during error-corrective training. We replicate the human result with a second cohort of learners in Appendix A.2. Notable is also the difference between shallow and deep linear networks. Response biases seem more transient in shallow networks and appear to more closely mirror human learners. However, quantitative comparisons are challenging due to inherently differing learning timescales.

6 Generic input correlations can equivalently drive OCS learning

We have established how the earliest phase of learning in linear networks is driven by the OCS. Crucially, in linear networks OCS learning hinges on bias terms in the network architecture. However,

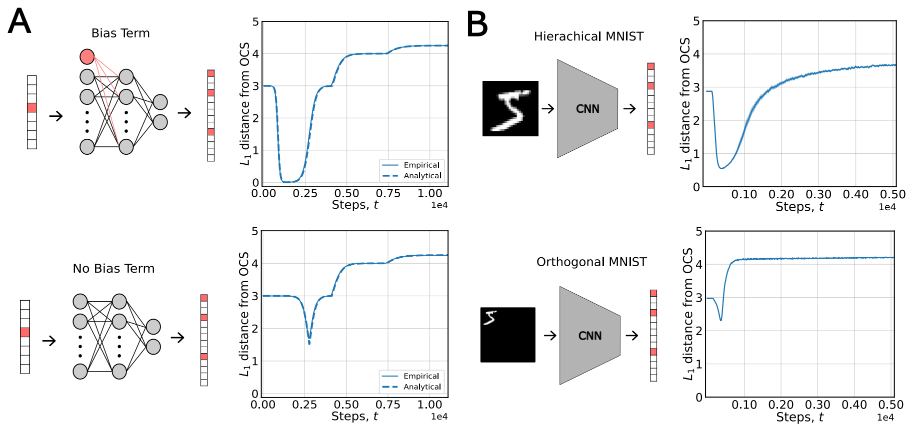


Figure 5: L_1 Distance from the OCS in Linear networks and CNNS. **A** Linear networks with (top) and without bias terms (bottom) trained on the hierarchical task **B** CNNs without bias terms trained on variants of the hierarchical task. *Top*: Normal inputs. *Bottom*: "orthogonal" image inputs which remove all between-class input correlations from the input data. Note, how CNNs do not learn the OCS in the absence of input correlations and bias terms.

in non-linear architectures, such as CNNs, the network is driven towards the OCS even in the absence of bias terms (Fig. 5B, Top). The appearance of the data term $\mathbf{X}^T\mathbf{X}$ in Proposition 3 suggests an equivalent effect that is induced by the data itself.

Corollary 1 (Input correlations induce early OCS)**.** *If $\mathbf{1}_N$ is an eigenvector of the data similarity matrix $\mathbf{X}^T\mathbf{X}$ with non-degenerate eigenvalue s_0, then the OCS response during early learning will be driven according to its magnitude.*

This statement follows directly from the joint diagonalisation of Eq. (1) and subsequent projection onto the OCS $\mathbf{y}_{ocs}$. We show a solvable case of OCS learning in linear networks under input correlations and in the absence of bias terms in Appendix A.11. We furthermore hypothesize that neural networks will be driven towards the OCS if training data contains more generic input correlations where $\mathbf{1}_N$ is not an exact eigenvector.

Setup. We trained CNNs on the hierarchical task in Section 4. Inputs were given by eight randomly sampled classes of MNIST (Fig. 5B, Top). To isolate the effect of input correlations we created a second dataset where randomly sampled classes of MNIST were copied on orthogonal subspaces of a larger image (Fig. 5B, bottom). Importantly, this procedure removes all between-class correlations.

Results. The main result of our experiment is displayed in Fig. 5. CNNs which learn from standard MNIST images are strongly driven towards the OCS. In contrast, early dynamics for the "orthogonal" MNIST do not display this tendency. Strikingly, the early dynamics with standard MNIST classes are highly similar to those observed in linear network with bias terms, while the dynamics for the latter task resemble those seen in the linear network without this feature. To verify that generic input correlations are indeed causing these differences we explore the eigenspectrum of the data correlation matrices. We sample 100 images from all 10 classes and compute a correlation matrix $\mathbf{X}^T\mathbf{X}$ from flattened images. First, we find that the eigenspectrum for standard MNIST images is dominated by a single eigenvector (Fig. 6, top-left). In contrast, the eigenspectrum of the orthogonal MNIST task does not display this property (Fig. 6, top-center). Further, recall that input bias terms lead to a non-degenerate constant eigenvector $\mathbf{1}_N$ in the input correlation matrix (Section 4). Similarly, we find that the first eigenvector $\mathbf{v}_1$ of $\mathbf{X}^T\mathbf{X}$ is indeed highly aligned to $\mathbf{1}_N$ (Fig. 6, right), whereas this is not the case in the orthogonal MNIST. We additionally show similar results for CIFAR-10 and CelebA. Theoretical considerations suggest that these correlations originate from an approximate symmetry in the data (see Appendix A.5.6).

Overall, we here demonstrate that early learning of the OCS can be driven by properties of the architecture (bias terms) or data (input correlations). Our results also highlight that input correlations are a common feature of standard image datasets: Early learning of the OCS might be a common occurrence when learning from such data. To see a practical implication of these results we briefly discuss fairness implications of OCS learning in Appendix A.10.

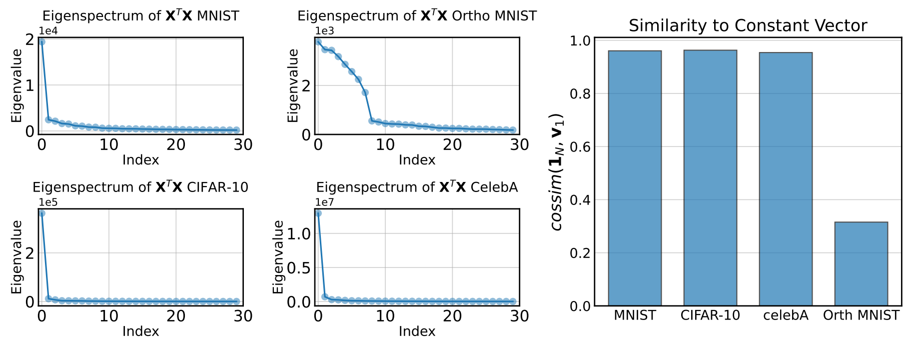


Figure 6: Dataset eigenspectra and constancy of first eigenvector for different image datasets. *Left:* Eigenspectra of $\mathbf{X}^T\mathbf{X}$ for different datasets. *Right:* Alignment of first eigenvector in $\mathbf{X}^T\mathbf{X}$ with the constant vector $\mathbf{1}_N$.

7 Discussion

In this work, we found that the inclusion of bias terms in linear networks shifts early learning towards the OCS, even when initialized at zero. We also highlight how OCS learning can equivalently be driven by input correlations. We demonstrated that early, input-independent simplicity biases occur in practice, affecting both non-linear networks and human learners. Our contribution complements prior work on simplicity biases by highlighting factors that drive networks in the *earliest* stages of learning; connecting input, output, and architecture. Overall, our findings highlight how simple linear networks can serve as useful tools to investigate simplicity biases in significantly more complex systems.

Relevance. We see promising applications for early OCS learning in the cognitive and behavioural sciences. OCS-like response biases have been noted previously (Herrnstein, 1961; Estes, 1964). However, we believe that a normative theory for these effects is still incomplete. Our theory identifies possible properties of the biological wetware or natural stimuli that may give rise to such biases.

While we do not study generalisation ourselves, we believe that OCS learning is practically important to understand *how* neural networks generalize or fail to generalize. Kang et al. (2024) has highlighted that networks will revert to OCS in a variety of generalisation settings. We demonstrate that the OCS component in the network function is acquired *early*, and is *retained* throughout training (effective singular values in Fig. 2 stay constant in late training). We believe that this retention of the OCS mode enables reversion.

OCS learning is also relevant when learning under class imbalance, a common problem in machine learning where datasets are frequently naturally imbalanced (Feldman, 2020; Van Horn and Perona, 2017), leading to a failure to learn information about minority classes (Ye et al., 2021). In Appendix A.10 we show an exactly solvable case of OCS learning in such settings and highlight how OCS learning can negatively impact performance for minority classes.

Limitations and future work. Our work is restricted to qualitative comparisons between linear networks and non-linear systems and our work only gives suggestive evidence of factors which drive early OCS learning in non-linear systems. We chose linear networks to allow for a rigorous description of the dynamics of learning. Methods from mean-field theory may provide a precise tool to analyze a wider range of systems directly. Second, the ambiguity between architecture and data in driving the OCS does not allow us to determine the underlying mechanism in human learners. Future studies might address this limitation by manipulating correlations in stimuli or by recording of neural data.

Reproduciblity statement. We provide the code to produce our simulation results in the supplementary material to this submission.

REFERENCES

Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. SGD on Neural Networks Learns Functions of Increasing Complexity. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the Spectral Bias of Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5301–5310. PMLR, May 2019. URL https://proceedings.mlr.press/v97/rahaman19a.html. ISSN: 2640-3498.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity Bias in Transformers and their Ability to Learn Sparse Boolean Functions, July 2023. URL http://arxiv.org/abs/2211.12316. arXiv:2211.12316 [cs].

Guillermo Valle-Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions, April 2019. URL http://arxiv.org/abs/1805.08522. arXiv:1805.08522 [cs, stat].

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3): 107–115, March 2021. ISSN 0001-0782, 1557-7317. doi: 10.1145/3446776. URL https://dl.acm.org/doi/10.1145/3446776.

Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. Neural networks trained with SGD learn distributions of increasing complexity. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28843–28863. PMLR, July 2023. URL https://proceedings.mlr.press/v202/refinetti23a.html. ISSN: 2640-3498.

Nora Belrose, Quintin Pope, Lucia Quirke, Alex Mallen, and Xiaoli Fern. Neural Networks Learn Statistics of Increasing Complexity, February 2024. URL http://arxiv.org/abs/2402.04362. arXiv:2402.04362 [cs].

Lukas Braun, Clémentine Dominé, James Fitzgerald, and Andrew Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Advances in Neural Information Processing Systems*, 35:6615–6629, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2b3bb2c95195130977a51b3bb251c40a-Abstract-Conference.html.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, February 2014. URL http://arxiv.org/abs/1312.6120. arXiv:1312.6120 [cond-mat, q-bio, stat].

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, June 2019. doi: 10.1073/pnas.1820226116. URL https://www.pnas.org/doi/full/10.1073/pnas.1820226116. Publisher: Proceedings of the National Academy of Sciences.

Timothy T. Rogers and James L. McClelland. *Semantic Cognition: A Parallel Distributed Processing Approach*. MIT Press, 2004. ISBN 978-0-262-18239-3. Google-Books-ID: AmB33Uz2MVAC.

David E. Rumelhart, James L. McClelland, and PDP Research Group. *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press, 1986. URL https://scholar.google.com/scholar?cluster=13839636846206420541&hl=en&oi=scholarr.

Katie Kang, Amrith Setlur, Claire Tomlin, and Sergey Levine. Deep Neural Networks Tend To Extrapolate Predictably, March 2024. URL http://arxiv.org/abs/2310.00873. arXiv:2310.00873 [cs].

R. J. Herrnstein. Relative and absolute strength of response as a function of frequency of reinforcement,. *Journal of the Experimental Analysis of Behavior*, 4(3):267–272, July 1961. ISSN 0022-5002. doi: 10.1901/jeab.1961.4-267. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1404074/.

William K. Estes. Probability Learning and Sequence learning. In Arthur W. Melton, editor, *Categories of Human Learning*, pages 89–128. Academic Press, January 1964. ISBN 978-1-4832-3145-7. doi: 10.1016/B978-1-4832-3145-7.50010-8. URL https://www.sciencedirect.com/science/article/pii/B9781483231457500108.

W. K. Estes and J. H. Straughan. Analysis of a verbal conditioning situation in terms of statistical learning theory. *Journal of Experimental Psychology*, 47(4):225–234, 1954. ISSN 0022-1015. doi: 10.1037/h0060989. URL https://doi.apa.org/doi/10.1037/h0060989.

Pete R. Jones, David R. Moore, Daniel E. Shub, and Sygal Amitay. The role of response bias in perceptual learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41 (5):1456–1470, September 2015. ISSN 1939-1285, 0278-7393. doi: 10.1037/xlm0000111. URL https://doi.apa.org/doi/10.1037/xlm0000111.

Joshua I. Gold, Chi-Tat Law, Patrick Connolly, and Sharath Bennur. The Relative Influences of Priors and Sensory Evidence on an Oculomotor Decision Variable During Perceptual Learning. *Journal of Neurophysiology*, 100(5):2653–2668, November 2008. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.90629.2008. URL https://www.physiology.org/doi/10.1152/jn.90629.2008.

William S. Verplanck, George H. Collier, and John W. Cotton. Nonindependence of successive responses in measurements of the visual threshold. *Journal of Experimental Psychology*, 44(4): 273–282, 1952. ISSN 0022-1015. doi: 10.1037/h0054948. URL https://doi.apa.org/doi/10.1037/h0054948.

James R. Hawker. The influence of training procedure and other task variables in paired-associate learning. *Journal of Verbal Learning and Verbal Behavior*, 3(1):70–76, February 1964. ISSN 0022-5371. doi: 10.1016/S0022-5371(64)80060-8. URL https://www.sciencedirect.com/science/article/pii/S0022537164800608.

Gordon H. Bower. An association model for response and training variables in paired-associate learning. *Psychological Review*, 69(1):34–53, January 1962. ISSN 1939-1471, 0033-295X. doi: 10.1037/h0039023. URL https://doi.apa.org/doi/10.1037/h0039023.

Jacob Feldman. Minimization of Boolean complexity in human concept learning. *Nature*, 407 (6804):630–633, October 2000. ISSN 0028-0836, 1476-4687. doi: 10.1038/35036586. URL https://www.nature.com/articles/35036586.

Noah D. Goodman, Joshua B. Tenenbaum, Jacob Feldman, and Thomas L. Griffiths. A Rational Analysis of Rule-Based Concept Learning. *Cognitive Science*, 32(1):108–154, 2008. ISSN 1551-6709. doi: 10.1080/03640210701802071. URL https://onlinelibrary.wiley.com/doi/abs/10.1080/03640210701802071. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1080/03640210701802071.

Nick Chater. Reconciling Simplicity and Likelihood Principles in Perceptual Organization. *Psychological review*, 103:566–81, July 1996. doi: 10.1037/0033-295X.103.3.566.

Tania Lombrozo. Simplicity and probability in causal explanation. *Cognitive Psychology*, 55(3): 232–257, November 2007. ISSN 0010-0285. doi: 10.1016/j.cogpsych.2006.09.006. URL https://www.sciencedirect.com/science/article/pii/S0010028506000739.

Jacob Feldman. The Simplicity Principle in Human Concept Learning. *Current Directions in Psychological Science*, 12(6):227–232, December 2003. ISSN 0963-7214. doi: 10.1046/j.0963-7214.2003.01267.x. URL https://doi.org/10.1046/j.0963-7214.2003.01267.x. Publisher: SAGE Publications Inc.

Kenji Fukumizu. Effect Of Batch Learning In Multilayer Neural Networks. June 1998.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90014-2. URL https://www.sciencedirect.com/science/article/pii/0893608089900142.

Andrew K. Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks, January 2019. URL http://arxiv.org/abs/1809.10374. arXiv:1809.10374 [cs, stat].

Samuel Liebana Garcia, Aeron Laffere, Chiara Toschi, Louisa Schilling, Jacek Podlaski, Matthias Fritsche, Peter Zatka-Haas, Yulong Li, Rafal Bogacz, Andrew Saxe, and Armin Lak. Striatal dopamine reflects individual long-term learning trajectories, December 2023. URL http://biorxiv.org/lookup/doi/10.1101/2023.12.14.571653.

Sygal Amitay, Yu-Xuan Zhang, Pete R. Jones, and David R. Moore. Perceptual learning: Top to bottom. *Vision Research*, 99:69–77, June 2014. ISSN 0042-6989. doi: 10.1016/j.visres.2013.11.006. URL https://www.sciencedirect.com/science/article/pii/S0042698913002800.

Anne E Urai, Jan Willem de Gee, Konstantinos Tsetsos, and Tobias H Donner. Choice history biases subsequent evidence accumulation. *eLife*, 8:e46331, July 2019. ISSN 2050-084X. doi: 10.7554/eLife.46331. URL https://doi.org/10.7554/eLife.46331. Publisher: eLife Sciences Publications, Ltd.

Yunshu Fan, Takahiro Doi, Joshua I. Gold, and Long Ding. Neural Representations of Post-Decision Accuracy and Reward Expectation in the Caudate Nucleus and Frontal Eye Field. *The Journal of Neuroscience*, 44(2):e0902232023, January 2024. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0902-23.2023. URL https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0902-23.2023.

Leo P. Sugrue, Greg S. Corrado, and William T. Newsome. Matching Behavior and the Representation of Value in the Parietal Cortex. *Science*, 304(5678):1782–1787, June 2004. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1094765. URL https://www.science.org/doi/10.1126/science.1094765.

Gilles Dutilh, Don van Ravenzwaaij, Sander Nieuwenhuis, Han L. J. van der Maas, Birte U. Forstmann, and Eric-Jan Wagenmakers. How to measure post-error slowing: A confound and a simple solution. *Journal of Mathematical Psychology*, 56(3):208–216, June 2012. ISSN 0022-2496. doi: 10.1016/j.jmp.2012.04.001. URL https://www.sciencedirect.com/science/article/pii/S0022249612000454.

Patrick Rabbitt and Bryan Rodgers. What does a Man do after he Makes an Error? An Analysis of Response Programming. *Quarterly Journal of Experimental Psychology*, 29(4): 727–743, November 1977. ISSN 0033-555X. doi: 10.1080/14640747708400645. URL http://journals.sagepub.com/doi/10.1080/14640747708400645.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. *ArXiv e-prints*, 2020. URL https://arxiv.org/abs/2002.02561. tex.eprint: 2002.02561.

Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022. Publisher: Elsevier tex.creationdate: 2022-07-20T21:54:34 tex.modificationdate: 2022-07-20T21:54:42.

Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A. Louis. Do deep neural networks have an inbuilt Occam's razor?, April 2023. URL http://arxiv.org/abs/2304.06670. arXiv:2304.06670 [cs, stat].

Li Deng. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6):141–142, November 2012. ISSN 1053-5888. doi: 10.1109/MSP.2012.2211477. URL http://ieeexplore.ieee.org/document/6296535/.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, March 2010. URL https://proceedings.mlr.press/v9/glorot10a.html. ISSN: 1938-7228.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf*, 2009.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild, September 2015. URL http://arxiv.org/abs/1411.7766. arXiv:1411.7766 [cs].

Oskar Perron. Zur Theorie der Matrices. *Mathematische Annalen*, 64(2):248–263, June 1907. ISSN 0025-5831, 1432-1807. doi: 10.1007/BF01449896. URL http://link.springer.com/10.1007/BF01449896.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, Chicago IL USA, June 2020. ACM. ISBN 978-1-4503-6979-4. doi: 10.1145/3357713.3384290. URL https://dl.acm.org/doi/10.1145/3357713.3384290.

Grant Van Horn and Pietro Perona. The Devil is in the Tails: Fine-grained Classification in the Wild, September 2017. URL http://arxiv.org/abs/1709.01450. arXiv:1709.01450 [cs].

Han-Jia Ye, De-Chuan Zhan, and Wei-Lun Chao. Procrustean Training for Imbalanced Deep Learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 92–102, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-66542-812-5. doi: 10.1109/ICCV48922.2021.00016. URL https://ieeexplore.ieee.org/document/9710650/.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/hash/5a4be1fa34e62bb8a6ec6b91d2462f5a-Abstract.html.

Daniel A Roberts, Sho Yaida, and Boris Hanin. The Principles of Deep Learning Theory. page 449.

Erich Hecke. Über orthogonal-invariante integralgleichungen. *Mathematische Annalen*, 78 (1):398–404, 1917. Publisher: Springer-Verlag tex.creationdate: 2022-07-23T15:10:14 tex.modificationdate: 2022-07-23T15:10:14.

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. In *International Conference on Machine Learning*, pages 2793–2802. PMLR, 2020. ISBN 2640-3498.

Ayşe Erzan and Aslı Tuncer. Explicit construction of the eigenvectors and eigenvalues of the graph Laplacian on the Cayley tree. *Linear Algebra and its Applications*, 586:111–129, February 2020. ISSN 0024-3795. doi: 10.1016/j.laa.2019.10.023. URL https://www.sciencedirect.com/science/article/pii/S002437951930463X.

Andries E. Brouwer and Willem H. Haemers. *Spectra of Graphs*. Springer Science & Business Media, December 2011. ISBN 978-1-4614-1939-6. Google-Books-ID: F98THwYgrXYC.

Haibo He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009. ISSN 1041-4347. doi: 10.1109/TKDE.2008.239. URL http://ieeexplore.ieee.org/document/5128907/.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.580. URL https://ieeexplore.ieee.org/document/7780949/.

Emanuele Francazi, Marco Baity-Jesi, and Aurelien Lucchi. A Theoretical Analysis of the Learning Dynamics under Class Imbalance, June 2023. URL http://arxiv.org/abs/2207.00391. arXiv:2207.00391 [cs, stat].

Bin Liu, Konstantinos Blekas, and Grigorios Tsoumakas. Multi-Label Sampling based on Local Label Imbalance, May 2020. URL `http://arxiv.org/abs/2005.03240`. arXiv:2005.03240 [cs].

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163: 3–16, September 2015. ISSN 0925-2312. doi: 10.1016/j.neucom.2014.08.091. URL `https://www.sciencedirect.com/science/article/pii/S0925231215004269`.

Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to Predict Visual Attributes in the Wild. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13013–13023, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-66544-509-2. doi: 10.1109/CVPR46437.2021.01282. URL `https://ieeexplore.ieee.org/document/9578060/`.

Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor W. Tsang. The Emerging Trends of Multi-Label Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7955–7974, November 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3119334. URL `https://ieeexplore.ieee.org/abstract/document/9568738`. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. pages 9268–9277, 2019. URL `https://openaccess.thecvf.com/content_CVPR_2019/html/Cui_Class-Balanced_Loss_Based_on_Effective_Number_of_Samples_CVPR_2019_paper.html`.

A Appendix / supplemental material

A.1 Overview

Our appendix has the following sections:

- In Appendix A.2, we describe the human experiment in more detail and show the results of a replication in a second cohort. We furthermore report statistical tests and describe ethical considerations.
- In Appendix A.3, we outline how we bring neural network and human responses displayed in Section 5 into a common space for direct comparison.
- In Appendix A.4, we outline how we compute the true negative rates, f^{tn} used in Section 4 and Section 5.
- In Appendix A.5, we provide additional theoretical derivations and remaining proofs to the statements in the main text.
- In Appendix A.6, we show OCS signatures in *shallow* networks with bias terms.
- In the short Appendix A.7, we show how linear networks without bias terms behave on the task in Section 5.
- In Appendix A.8, we describe hyperparameters, datasets, and further training details used for our CNN experiments.
- In Appendix A.9, we describe the results of additional experiments investigating early emergence of the OCS in non-linear models.
- In Appendix A.10 we show an additional solvable case of linear networks with bias terms under class imbalance.
- In Appendix A.11 we show OCS learning in linear networks with input correlations but in the absence of bias terms.
- In Appendix A.12 we discuss additional connections of our work to multi-label learning.

A.2 Human learning experiment

We directly translated the hierarchical task setup used by Saxe et al. (2019) into an experimental paradigm. Our design attempts to stay as close to the original task structure used for neural networks as possible. We designed the task as a mapping from 8 distinct input stimuli represented as planets to a set of 3 associated output stimuli represented as plants (see Fig. 7, left).

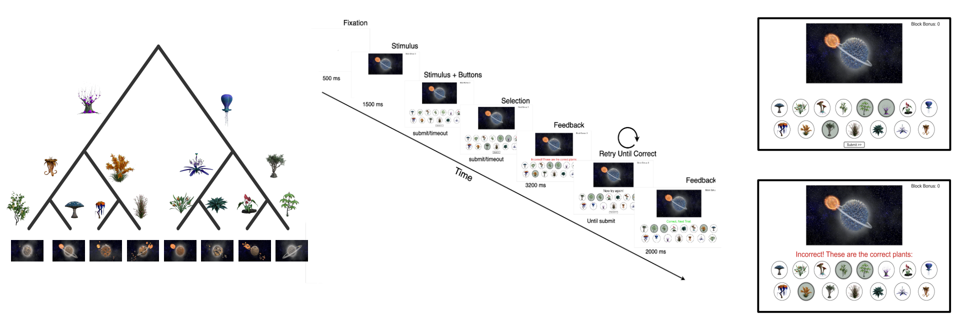


Figure 7: Human task design. *Left:* Hierarchical learning task, adapted for human participants. *Centre:* Trial structure as experienced by human participants. *Right:* Example screen during response period (top), Example screen during feedback period (bottom).

In the task, participants had to learn to associate which outputs properties are associated with each input. Unbeknownst to the participants we imposed a hierarchical structure on output targets (Fig. 7, left). In the structure some output labels are associated with more than one input. As a control for analyses we also included an additional control input-output pair (similarly represented by a planet and a plant; not shown here and excluded from current analysis). We recruited a cohort of 10 subjects that were trained over the course of three days with one daily session. The cohort was recruited as part of a larger neuroimaging experiment but our analysis presented here is exclusive to behavioural results. We further replicated our results in a second cohort of 46 human subjects recruited via the online platform Prolific (prolific.com). Results of the replication of the study can be seen in Fig. 8.

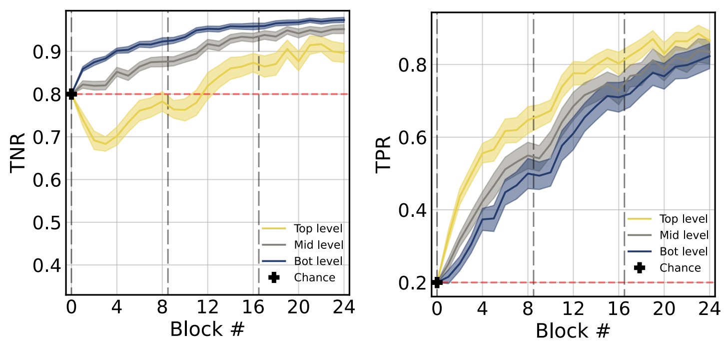
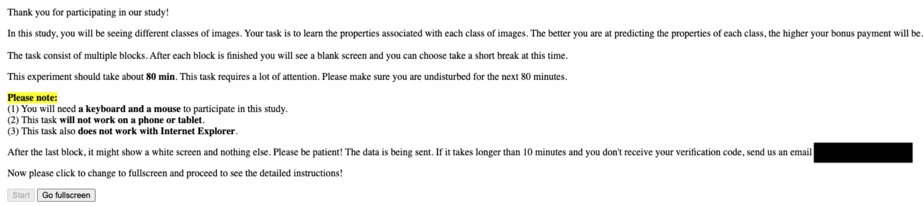


Figure 8: The human replication cohort. While learning is slower, the qualitative pattern indicating reliance on the OCS is replicated. *Left*: TNR rate. *Right:* TPR rate.

Each day of at home training consisted of 8 blocks of training with 22 trials each (160 standard trials and 16 control trials) which lasted about one hour. The trial structure during training is shown in Fig. 7, centre. During training trials, subjects were shown the stimulus on screen and were required to press three buttons, presented below the planet image (Fig. 7, right). The subjects received fully informative feedback on each trial and were forced to repeat the trial in the case of incorrect selection until the correct properties were selected. The location of buttons was shuffled on screen for each trial and for each forced repetition. For each button clicked correctly on their first attempt participants received a bonus point. We displayed a block-wise bonus in the corner of the screen throughout the task. Participants were payed slightly above local minimum wage as a baseline and received a substantial performance dependent bonus (on average about one-third of the baseline pay). We include a screenshot of the initial instructions in Fig. 9. Beyond this initial instruction screen participants received more nuanced instructions about clicking of buttons and feedback in the beginning of the task.

Statistical tests. While our focus is on qualitative patterns in human behaviour, we compute statistical tests on the true negative rates for human results seen in the main text (Fig. 4). We averaged all blocks in a given day and performed a two-way repeated measures ANOVA to assess the effect of day and hierarchy level on true negative rates. The two-way repeated measures ANOVA revealed significant main effects of day $F(2, 18) = 57.22,\ p < .0001,\ \eta^2 = .25$ and level $F(2, 18) = 6.25,\ p = .033,\ \eta^2 = .18$. Beyond this we also found a significant interaction of day and level $F(4, 36) = 9.795,\ p = .0056,\ \eta^2 = .042$. A Mauchly test indicated that the assumption of sphericity had been violated for level $\chi^2 = .03,\ p < .5$ and the interaction term $\chi^2 = .006,\ p < .5$. Significance values are reported with Greenhouse-Geisser correction. The results confirm that performance between levels are significantly different depending on day and hierarchical level.

Ethical considerations. Human participants performed a simple, computerised learning task without the collection of personal identifiable information or substantial deception. Human data collection was handled strictly in line with institutional guidelines and under institutional review board approval. We obtained informed consent for each participant before commencing the study. We highlighted that participants could withdraw at any time without penalty or loss of compensation by simply exiting full-screen or informing the experimenter. We provided contact emails in the case of concern or questions. Data was handled in a strictly anonymised format and stored on password secured devices. Participants were payed above minimum wage for their country of origin.


Thank you for participating in our study!

In this study, you will be seeing different classes of images. Your task is to learn the properties associated with each class of images. The better you are at predicting the properties of each class, the higher your bonus payment will be.

The task consist of multiple blocks. After each block is finished you will see a blank screen and you can choose take a short break at this time.

This experiment should take about **80 min**. This task requires a lot of attention. Please make sure you are undisturbed for the next 80 minutes.

Please note:
(1) You will need **a keyboard and a mouse** to participate in this study.
(2) This task **will not work on a phone or tablet**.
(3) This task also **does not work with Internet Explorer**.

After the last block, it might show a white screen and nothing else. Please be patient! The data is being sent. If it takes longer than 10 minutes and you don't receive your verification code, send us an email [redacted]

Now please click to change to fullscreen and proceed to see the detailed instructions!

Start Go fullscreen

Figure 9: Initial instructions received by participants after the collection of informed consent.

A.3 Discretizing network responses for comparison to humans

Discretization. In our task, neural networks produced continuous outputs. This is distinct from human learners who were required to give discrete responses. We now describe the discretization that allows us to compare human and network responses. Fundamentally, we conceptualise inference as a noisy process by which responses are sampled from a distribution over output labels. That is, we treat outputs from our linear network as logits. We first feed network outputs $\hat{\mathbf{y}}_i$ through a softmax function with temperature 0.2 and subsequently sample three responses without replacement. The procedure maps continuous outputs $\hat{\mathbf{y}}_i$ to binary responses vectors in $\{0,1\}^{N_{out}}$.

Expected solutions. Here we describe the derivation of expected solutions used in Fig. 4, dashed lines. The derivation of these "expected responses" under the sampling procedure allows to make the reliance of network responses on the exact solutions in Section 3 clear.

Consider network outputs $\hat{\mathbf{y}}_i(t) = \mathbf{W}^2(t)\mathbf{W}^1(t)\mathbf{x}_i$. We transform these outputs through a softmax function $\sigma_\beta : \mathbb{R}^{N_{out}} \to (0,1)^{N_{out}}$. Let $S = \{s_1, s_2, s_3\}$ denote the set of three unique response indices sampled from $\sigma_\beta(\hat{\mathbf{y}}_i(t))$ without replacement, where $s_n \in \{1, 2, \ldots, N_{out}\}$ for $n = 1, 2, 3$, and all s_n are hence distinct. The probability distribution $\sigma_\beta(\hat{\mathbf{y}}_i(t))$ is dependent on time t, therefore denote the produced probability of S as $P_t(S)$. For each of these sets S we can compute an associated true negative rate for each of the $k \in \{1, 2, 3\}$ levels in the hierarchy. We denote this random variable as X_S^k. We can then compute expected solutions to inference behaviour as

$$\mathbb{E}_t[X_S^k] = \sum_{S \subseteq \{1,2,\ldots,m\},\, |S|=3} P_t(S) X_S^k \tag{6}$$

A.4 True Negative Rates

Here we describe the metric used in the bottom panel of Fig. 3 and in Fig. 4. The metrics effectively describes true negative rates (correct-rejection scores). We use the metric on continuous network responses in $\mathbb{R}^{N_{out}}$ in Fig. 3. We also use the metric on discretised networks responses in $\{0,1\}^{N_{out}}$ and for human responses in $\{0,1\}^{N_{out}}$ in Fig. 4.

Given responses $\hat{\mathbf{y}}$ and target vectors $\mathbf{y} \in \mathbb{R}^{N_{out}}$ the metric computes the alignment between target and response vectors while only focusing on zero entries in $\mathbf{y}$. Furthermore we compute the metric separately for the $k \in \{1, 2, 3\}$ separate levels of the hierarchy where the entries s_k and e_k denote relevant start and end indices of level k in the vectors $\hat{\mathbf{y}}$ and $\mathbf{y}$. The metric is then computed as

$$f_k^{tn}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{(\mathbf{1}_{N_{out}} - \hat{\mathbf{y}})_{s_k:e_k}^T (\mathbf{1}_{N_{out}} - \mathbf{y})_{s_k:e_k}}{(\mathbf{1}_{N_{out}} - \mathbf{y})_{s_k:e_k} (\mathbf{1}_{N_{out}} - \mathbf{y})_{s_k:e_k}}, \tag{7}$$

where $s_k : e_k$ is a "slicing" notation that takes the subvector between indices s_k and e_k.

If for all desired entries of 0 in $\mathbf{y}$ the vector $\hat{\mathbf{y}}$ is equal to 0 the metric will be at 1. Correspondingly if entries in $\hat{\mathbf{y}}$ are larger than zero the metric $f_k^{tn}(\hat{\mathbf{y}}, \mathbf{y})$ will decrease. Thus, the metric measures wrong beliefs about the presence of target labels across the different levels of the hierarchy.

A.5 Additional theoretical results and proofs

A.5.1 Equivalence of bias terms

In this section, we give more detail on the method used in in Section 3 of how to reformulate a bias term in terms of the network weights and a constant feature in the input.

Consider a network with an explicit input bias term $\mathbf{b}^1$,

$$\hat{\mathbf{y}} = \tilde{\mathbf{W}}^1 \tilde{\mathbf{x}} + \tilde{\mathbf{b}}^1$$

This is equivalent to introducing a constant component to the vector $\mathbf{x}$,

$$\tilde{\mathbf{x}} \rightarrow \mathbf{x} := \begin{bmatrix} 1 \\ \tilde{\mathbf{x}} \end{bmatrix},$$

and using the network

$$\hat{\mathbf{y}} = \mathbf{W}^1 \mathbf{x},$$

as we can write

$$\begin{aligned}
\left(\mathbf{W^1 x}\right)_m &= \sum_{j=0}^{N_{in}} W^1_{mj} x_j \\
&= W^1_{m0} 1 + \sum_{j=1}^{N_{in}} W^1_{mj} x_j \\
&= W^1_{m0} 1 + \sum_{j=0}^{N_{in}-1} \tilde{W}^1_{mj} \tilde{x}_j \\
&\equiv b^1_m + \sum_{j=0}^{N_{in}-1} \tilde{W}^1_{mj} \tilde{x}_j .
\end{aligned}$$

In order to match a given i.i.d. initialization $b^1_m \sim \mathcal{N}\left(0,\, \sigma_b^2\right)$ where $\sigma_b \neq \sigma_w$, the component that needs to be added to $\tilde{\mathbf{x}}$ to get equivalence needs to be σ_b / σ_w.

A.5.2 Learning dynamics for bias terms

We here derive analytical expressions for the learning speeds of input and output bias terms for a two-layer deep linear network discussed in the main text,

$$\hat{\mathbf{y}} = \mathbf{W}^2 \left(\mathbf{W}^1 \mathbf{x} + \mathbf{b}^1\right) + \mathbf{b}^2 .$$

We decompose $\mathbf{W}^2 = \mathbf{U}\mathbf{A}^{(2)}\mathbf{R}^{(2)}$ and $\mathbf{W}^1 = \mathbf{R}^{(1)}\mathbf{A}^{(1)}\mathbf{V}$ by means of a singular value decomposition (SVD). We here make the assumption of balancedness $\mathbf{W}^1(0)\mathbf{W}^{1T}(0) = \mathbf{W}^{2T}(0)\mathbf{W}^2(0)$ (Braun et al., 2022) at the beginning of training, which implies $\mathbf{R}^{(2)}\mathbf{S}^{(2)2}\mathbf{R}^{(2)T} = \mathbf{R}^{(1)}\mathbf{S}^{(1)2}\mathbf{R}^{(1)T}$. For clarity, we further assume the simplification

$$\mathbf{R}^{(2)T} = \mathbf{R}^{(1)} =: \mathbf{R},\ \mathbf{A}^{(2)} = \mathbf{A}^{(1)} =: \sqrt{\mathbf{A}}.$$

We here just state these relations without further comment to complement the respective derivation for the weights in (Saxe et al., 2014). This decomposition then allows to rewrite the gradients.

Input bias term

$$
\begin{aligned}
\tau \frac{d}{dt}\mathbf{b}^1 &= \nabla_{\mathbf{b}^1}\mathcal{L} \\
&= (\mathbf{y}-\hat{\mathbf{y}})^T \mathbf{W}^2 \\
&= \left(\mathbf{y}-\left(\mathbf{W}^2\left(\mathbf{W}^1\mathbf{x}+\mathbf{b}^1\right)+\mathbf{b}^2\right)\right)^T \mathbf{W}^2 \\
\mathbb{E}_{\mathbf{x}} &\rightarrow \left(\bar{\mathbf{y}}-\mathbf{W}^2\left(\mathbf{W}^1\bar{\mathbf{x}}+\mathbf{b}^1\right)-\mathbf{b}^2\right)^T \mathbf{W}^2 \\
&= \left(\bar{\mathbf{y}}-\mathbf{UAV}\bar{\mathbf{x}}-\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}\mathbf{b}^1-\mathbf{b}^2\right)^T \mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^T \\
&= \bar{\mathbf{y}}^T\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^T - \bar{\mathbf{x}}^T\mathbf{V}^T\mathbf{A}\mathbf{R}^T - \mathbf{b}^{1T}\mathbf{RAR} - \mathbf{b}^{2T}\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^T \\
&= (\mathbf{Y}\mathbf{1}_N)^T\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^T - (\mathbf{X}\mathbf{1}_N)^T\mathbf{V}^T\mathbf{AR} - \mathbf{b}^{1T}\mathbf{RAR}^T - \mathbf{b}^{2T}\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^T.
\end{aligned}
$$

Here, we denoted the expectation over the data samples as $\mathbb{E}_{\mathbf{x}}$. Projecting from the right with $\mathbf{R}_\alpha \in \mathbb{R}^{N_{\text{hidden}}}$ gives

$$
\tau \frac{d}{dt}\left(\mathbf{b}^{1T}\mathbf{R}_\alpha\right) = \bar{\mathbf{y}}^T\mathbf{U}_\alpha\sqrt{a_\alpha} - \bar{\mathbf{x}}^T\mathbf{V}_\alpha^T a_\alpha - \mathbf{b}^{1T}\mathbf{R}_\alpha a_\alpha - \mathbf{b}^{2T}\mathbf{U}_\alpha\sqrt{a_\alpha}. \tag{8}
$$

Output bias term

$$
\begin{aligned}
\tau \frac{d}{dt}\mathbf{b}^2 &= \left(\mathbf{y}-\left(\mathbf{W}^2\left(\mathbf{W}^1\mathbf{x}+\mathbf{b}^1\right)+\mathbf{b}^2\right)\right) \\
\mathbb{E}_{\mathbf{x}} &\rightarrow \bar{\mathbf{y}}-\mathbf{W}^2\left(\mathbf{W}^1\bar{\mathbf{x}}+\mathbf{b}^1\right)-\mathbf{b}^2 \\
&= \bar{\mathbf{y}}-\mathbf{UAV}\bar{\mathbf{x}}-\mathbf{U}\sqrt{\mathbf{A}}\mathbf{R}^{\mathbf{T}}\mathbf{b}^1-\mathbf{b}^2 \\
&= \mathbf{Y}\mathbf{1}_N-\mathbf{UAVX}\mathbf{1}_N-\mathbf{U}\mathbf{R}^{\mathbf{T}}\mathbf{b}^1-\mathbf{b}^2.
\end{aligned} \tag{9}
$$

Notably, the derivative in Eq. (8) is proportional to the singular vectors of the weights a_α, so that its growth is attenuated, analogous to the sigmoidal growth in deep linear networks (Saxe et al., 2014). In contrast, the learning signal $\frac{d}{dt}\mathbf{b}^2$ in Eq. (9) is not affected by the initialization of the weights and is hence $\mathcal{O}(1)$ already at the beginning of learning, reminiscent of shallow networks.

A.5.3 Integrated formulation of architectural biases.

In the main text, we have analysed how bias terms on the *input* layer affect the singular value spectrum. Our empirical results in Section 4.1 suggest a more general dynamical bias towards the OCS stemming purely from architectural properties. Here, we use the neural tangent kernel $\mathsf{NTK}(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_k \frac{d\hat{\mathbf{y}}_i}{d\theta_k}\frac{d\hat{\mathbf{y}}_{i'}^T}{d\theta_k}$ (Jacot et al., 2018) to directly and comprehensively describe the affected time evolution of the network response $\frac{d}{dt}\hat{\mathbf{y}}_i = \mathsf{NTK}(\mathbf{y}_i - \hat{\mathbf{y}}_i)$ at the cost of a closed-form solution. Because changes in network outputs are proportional to the NTK it can been viewed as an architecture-induced learning rate (Roberts et al.). For a review and derivation of the NTK, see Appendix A.5.8. For completeness, we now consider a network that contains input $\mathbf{b}^1$ and output $\mathbf{b}^2$ bias terms.

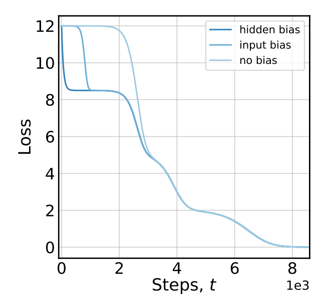


Figure 10: Loss curves for different bias variations.

Proposition 3 (NTK of linear networks with bias terms). *Consider a two-layer linear network with input and output-layer bias $\hat{\mathbf{Y}} = \mathbf{W}^2(\mathbf{W}^1\mathbf{X}+\mathbf{b}^1)+\mathbf{b}^2$ in the high-dimensional regime. Furthermore, assume weights are initialized i.i.d. $W_{ij}^\ell \sim \mathcal{N}(0, \sigma^2_{\mathbf{W}^\ell}/N_{in}^\ell)$ in each layer. Then, the neural tangent kernel of in early training in expectation $\mathbb{E}_{\mathbf{W}}$ reads*

$$
\mathsf{NTK}(\mathbf{X},\mathbf{X}) = \sigma^2_{\mathbf{W}^2}\mathbf{I}_{N_{out}} \otimes \left(2\mathbf{X}^T\mathbf{X} + \underbrace{\mathbf{1}_N\mathbf{1}_N^T}_{\leftrightarrow \mathbf{b}^1}\right) + \mathbf{1}_{N_{out}}\mathbf{1}_{N_{out}}^T \otimes \underbrace{\mathbf{1}_N\mathbf{1}_N^T}_{\leftrightarrow \mathbf{b}^2}. \tag{10}
$$

The tensor product $\otimes$ separates the components that operate on output and sample space. We briefly review the NTK and derive this expression in the next section. The highlighted terms originate from the bias term $\frac{d\hat{\mathbf{y}}_i}{d\mathbf{b}}\frac{d\hat{\mathbf{y}}_{i'}^T}{d\mathbf{b}}$ entering the NTK, manifesting in the appearance of the constant mode $\mathbf{1}_N$. Importantly, these terms do not scale with the size of the learned bias – they are present even if the bias is initialized at zero. Intuitively, their contribution stems from the architecture's *potential* to learn a bias, enabling rapid changes in output $\hat{\mathbf{y}}$. The NTK also reveals a qualitative difference between input and output bias: Whereas the term that is induced by $\mathbf{b}^1$ shows attenuated growth due to the multiplication by the weights of initial scale $\sigma_{\mathbf{W}^2} \ll 1$, the output bias $\mathbf{b}^2$ immediately changes the output significantly. Loss curves which demonstrate the effect of different bias terms are displayed in Fig. 10.

PROOFS

A.5.4 FEASIBILITY OF CLOSED-FORM SOLUTION

Proposition 1 (Feasibility of closed-form learning dynamics)**.** *For any input data $\mathbf{X} \in \mathbb{R}^{N_{in} \times N}$ and output data $\mathbf{Y} \in \mathbb{R}^{N_{out} \times N}$ it is possible to diagonalize $\mathbf{\Sigma}^x$ by the right singular vectors $\mathbf{V}$ of $\mathbf{\Sigma}^{yx}$ if $\mathbf{Y}^T\mathbf{Y}$ and $\mathbf{X}^T\mathbf{X}$ commute. The converse holds true only if $\mathbf{X}$ has a left inverse.*

Proof. We would like to know when the right singular vectors $\mathbf{V}$ (denote as $\mathbf{V}^{yx}$ here for clarity) of $\mathbf{\Sigma}^{yx} = \mathbf{U}^{yx}\mathbf{S}^{yx}\mathbf{V}^{yx}$ match these of $\mathbf{\Sigma}^x = \mathbf{U}^x\mathbf{S}^x\mathbf{V}^x$. First, to reduce the problem to $\mathbf{V}^{yx}$, note that $\mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx} = \mathbf{V}^{yx}\mathbf{S}^{yx2}\mathbf{V}^{yx}$, so that what remains to show is $\left[\mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx}, \mathbf{\Sigma}^{xx}\right] = 0$, where $[\mathbf{A}, \mathbf{B}] := \mathbf{AB} - \mathbf{BA}$ denotes the commutator between two matrices $\mathbf{A}$ and $\mathbf{B}$. We compute the two terms as

$$\mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx}\mathbf{\Sigma}^{xx} = \mathbf{X}\mathbf{Y}^T\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{X}^T$$
$$\mathbf{\Sigma}^{xxT}\mathbf{\Sigma}^{yxT}\mathbf{\Sigma}^{yx} = \mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{Y}^T\mathbf{Y}\mathbf{X}^T$$

The commutator vanishes if these terms match, which happens for the simpler equality

$$\mathbf{Y}^T\mathbf{Y}\mathbf{X}^T\mathbf{X} = \mathbf{X}^T\mathbf{X}\mathbf{Y}^T\mathbf{Y},$$

or $\left[\mathbf{Y}^T\mathbf{Y}, \mathbf{X}^T\mathbf{X}\right] = 0$. The converse follows only if the transformation $\mathbf{X}\ldots\mathbf{X}^T$ in the former equation is invertible, which is the case if a left inverse $\mathbf{X}^{-1}\mathbf{X} = \mathbf{I}_{N_{in}}$ exists.

A.5.5 OCS AND SHARED PROPERTIES CORRESPOND TO EACH OTHER

We here link the OCS and shared properties stand in close relation, as the eigenvector $\mathbf{1}_N$ represents properties that are shared across all data samples.

Proposition 2 (The OCS is linked to shared properties)**.** *If $\mathbf{1}_N$ is an eigenvector to the similarity matrix $\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{N \times N}$, then the sample-average $\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i$ will be an eigenvector to the correlation matrix $\mathbf{X}\mathbf{X}^T \in \mathbb{R}^{N_{in} \times N_{in}}$ with identical eigenvalue λ. An analogous statement applies for $\mathbf{Y}^T\mathbf{Y}$ and $\mathbf{Y}\mathbf{Y}^T$. The converse does not hold true in general.*

Proof.

$$\mathbf{X}\mathbf{X}^T\bar{\mathbf{x}} = \left(\mathbf{X}\mathbf{X}^T\right)\frac{1}{N}\mathbf{X}\mathbf{1}_N = \frac{1}{N}\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)\mathbf{1}_N = \frac{1}{N}\mathbf{X}\lambda\,\mathbf{1}_N = \lambda\frac{1}{N}\mathbf{X}\mathbf{1}_N = \lambda\bar{\mathbf{x}}.$$

A.5.6 CONSTANT DATA MODE $\mathbf{1}_N$ IS RELATED TO SYMMETRY IN THE DATA

This section gives proof sketches based on symmetry in the dataset that are sufficient to make $\mathbf{1}_N$ an eigenvector to $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$, and in particular hold for the dataset that we are considering. We anticipate that it is possible to formulate these statements in a more universal way by fully leveraging the cited literature.

The assumptions on symmetry should intuitively at least hold in an approximate manner for many datasets, we expect that they indeed are the reason why we observe a prevalence of $\mathbf{1}_N$, although they are not a necessary condition.

Continuously supported data $\mathbf{x} \in \mathbb{R}^{N_{in}}$

Proposition 4 (Continuous symmetry induces $\mathbf{1}_N$)**.** *If the pairwise correlations $\mathbf{y}_i^T \mathbf{y}_{i'}$ in a dataset are rotationally symmetric, its similarity matrix $\mathbf{Y}^T\mathbf{Y}$ has eigenvector $\mathbf{1}_N$. Note that this is a weaker assumption than the data itself being symmetric.*

Proof. We assume that $\mathbf{Y}, \mathbf{X}$ have been sampled from a ground truth data distribution $p(\mathbf{y}, \mathbf{x})$. If $p(\mathbf{y})$ is rotationally symmetric and $\mathbf{X}$ is comprised of samples $\mathbf{x}$ that are uniformly distributed on the hypersphere, we can introduce the kernel function $\mathbf{y}_{\mathbf{x}_i}^T \mathbf{y}_{\mathbf{x}_{i'}} = k(\mathbf{x}_i,\, \mathbf{x}_{i'}) = k(\mathbf{R}_l \mathbf{x}_i,\, \mathbf{R}_l \mathbf{x}_{i'}) = k(\mathbf{x}_i^T \mathbf{x}_{i'})$ for any R_l that is a representation of the group of rotations $G = \mathrm{SO}(N_{in})$ that faithfully acts on the "subsampled" hypersphere $\mathbf{X}$ comprised of vectors $\mathbf{x} \in \mathbb{R}^{N_{in}}$. It therefore only depends on the pairwise input similarity (hence sometimes called dot-product kernel). If follows that for all vectors $\mathbf{v}(\mathbf{X}) \in \mathbb{R}^N$ that are evaluations of the functions of the sample points $\mathbf{X}$

$$\mathbf{Y}^T\mathbf{Y}\,\mathbf{v} = k(\mathbf{X}^T\mathbf{X})\,\mathbf{v} = k((\mathbf{R}_l\mathbf{X})^T\,(\mathbf{R}_l\mathbf{X}))\,\mathbf{v} = \mathbf{R}_k^T k(\mathbf{X}^T\mathbf{X})\mathbf{R}_l\,\mathbf{v} \;\Leftrightarrow\; \left[\mathbf{Y}^T\mathbf{Y},\, \mathbf{R}_l\right] = 0,$$

where $[\mathbf{A}, \mathbf{B}] =: \mathbf{AB} - \mathbf{BA}$ is the commutator between two matrices.

It follows that we must have for all rotations $\mathbf{R}_l$

$$\mathbf{R}_l\left(\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N\right) = \mathbf{Y}^T\mathbf{Y}\mathbf{R}_l\,\mathbf{1}_N = \mathbf{Y}^T\mathbf{Y}\,\lambda_{\mathbf{R}_l}\mathbf{1}_N = \lambda_{\mathbf{R}_l}\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N$$

with eigenvalue $\lambda_{\mathbf{R}_l} = 1$.

meaning that $\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N$ is an eigenvector to *all* $\mathbf{R}_l$. This can only be the case if $\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N \propto \mathbf{1}_N$, as this is the only vector of values on the sphere that is invariant under any rotations.

We point out that it can be shown more generally with tools from functional analysis that the full spectrum of this kernel operator k are the spherical harmonics if the data measure $p(\mathbf{x})$ is spherically symmetric (Hecke, 1917), see (Dutordoir et al., 2020) for a modern presentation with tools from calculus. As the first harmonic $\mathcal{Y}_{l=0,m=0}(\mathbf{x})$ is constant, it follows that also the constant function $\mathbf{1}(\mathbf{x}) \equiv 1$ is an eigenfunction when drawing a finite set of samples from this kernel.

Data on a graph $\mathbf{x} \in \mathbb{R}^{N_{in}}$ We here prove that the former statement holds for the hierarchical dataset that is discussed in the main text, i.e. that $\mathbf{1}_N$ is an eigenvector to $\mathbf{Y}^T\mathbf{Y}$.

First, note that it is easy to convince oneself of this by writing down the matrices explicitly: Then, as the rows are just permutations of one another, $\mathbf{1}_N$ is immediately identified as an eigenvector, because $\sum_{i'}^N \mathbf{Y}_i^T \mathbf{Y}_{i'} 1_{i'} = \mathbf{Y}_i^T \left(\sum_{i'} \mathbf{Y}_{i'}\right)$ will then not depend on i and hence be proportional to $\mathbf{1}_N$.

To connect with the former symmetry-based argument Appendix A.5.6, we here however give a proof that is based on the symmetry in the data:

Proposition 5 (Discrete symmetry induces $\mathbf{1}_N$)**.** *Consider a connected Cayley tree graph with adjacency matrix $\mathbf{A}$ and nodes $\mathbf{x}_i$. Furthermore, let $\mathbf{R}_l \in G$ be an element of a faithful representation of the symmetry group G that acts on the graph nodes $\mathbf{v}$, i.e. that leaves its adjacency matrix invariant, $[\mathbf{R}_l, \mathbf{A}] = 0\,\forall\,\mathbf{R}_l$.*

If $\mathbf{Y}$ are labels associated with the leaf nodes $\mathbf{X}$ (the outermost generation of the graph, see (Erzan and Tuncer, 2020)) and there exists a similarity function k such that $\mathbf{y}_{\mathbf{x}_i}^T \mathbf{y}_{\mathbf{x}_{i'}} = \mathbf{y}_{\mathbf{R}_l\mathbf{x}_i}^T \mathbf{y}_{\mathbf{R}_l\mathbf{x}_{i'}} = k(d(\mathbf{x}_i, \mathbf{x}_{i'}))\,\forall\,\mathbf{R}_l$ where d is the geodesic distance on the graph, $\mathbf{1}_N$ will be an eigenvector of $\mathbf{Y}^T\mathbf{Y}$.

Proof. From the symmetry assumption on the labels, we again have for any vector $\mathbf{v}$ of node loadings $\left[\mathbf{R}_l,\, \mathbf{Y}^T\mathbf{Y}\right]\mathbf{v} = 0\,\forall\,\mathbf{R}_l \in G$. From this, we find that

$$\mathbf{R}_l\left(\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N\right) = \mathbf{Y}^T\mathbf{Y}\,\mathbf{R}_l\mathbf{1}_N = \mathbf{Y}^T\mathbf{Y}\,\lambda_{\mathbf{R}_l}\mathbf{1}_N = \lambda_{\mathbf{R}_l}\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N\,\forall\,\mathbf{R}_l$$

This shows that $\mathbf{Y}^T\mathbf{Y}\,\mathbf{1}_N$ is an eigenvector of $\mathbf{R}_l$ with eigenvalue $\lambda_{\mathbf{R}_l} = 1$ for any element of the symmetry group. The only vector $\mathbf{v}$ that is invariant under *all* symmetry operations of the graph is the constant vector $\mathbf{1}_N$.

We briefly point out the rich literature on spectral graph theory (for example (Brouwer and Haemers, 2011; Erzan and Tuncer, 2020)) that might allow making statements about the nature of the eigenvalues and other eigenvectors as a function of the graph topology. We expect that this is possible because the literature in the continuous case discussed in the next paragraph bases their arguments on the Laplacian on the sphere, an operator that can be extended to graphs as well. We leave these exploration for future work.

Corollary 2. *Because $k(\mathbf{X}^T\mathbf{X}) := \mathbf{X}^T\mathbf{X}$ defines a particular case of input-output similarity mapping, $\mathbf{1}_N$ is also an eigenvector to $\mathbf{X}^T\mathbf{X}$ under the former assumptions of uniform data distribution.*

A.5.7 Constant data mode $\mathbf{1}_N$ is the leading eigenvector

Here, we prove that the constant eigenvector $\mathbf{1}_N$ which is responsible for the OCS solution is associated with the *leading* eigenvalue of the input-output correlation matrix and hence drives early learning.

Theorem 1 (Early learning is biased by the OCS mode)**.** *If $\mathbf{1}_N$ is a joint non-degenerate eigenvector to positive input and output similarity matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$, the OCS mode $s_{ocs}\bar{\mathbf{y}}\bar{\mathbf{x}}^T$ will have leading spectral weight $s_0 \equiv s_{ocs}$ in the SVD of the input-output correlation matrix $\mathbf{\Sigma}^{yx}$.*

Proof. Let $\mathbf{1}_N$ be an eigenvector to both similarity matrices $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$ associated with eigenvalue $\tilde{\lambda}$. Moreover, let $\mathbf{X}^T\mathbf{X}$ and $\mathbf{Y}^T\mathbf{Y}$ have positive entries. Then, the Perron-Frobenius theorem (Perron, 1907) guarantees that $\tilde{\lambda}$ is indeed the leading eigenvector to $\mathbf{Y}^T\mathbf{Y}$, $\tilde{\lambda} \equiv \lambda_0 = s_0^2$.

By Proposition 2, $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ are now also the leading eigenvectors for $\mathbf{X}\mathbf{X}^T$ and $\mathbf{Y}\mathbf{Y}^T$. Because the eigenvectors of $\mathbf{Y}\mathbf{Y}^T$ and $\mathbf{X}\mathbf{X}^T$ are the left and right singular vectors of $\mathbf{\Sigma}^{yx}$, respectively, with the eigenvalues being the squares of the singular values, it follows that

$$s_0\mathbf{u}_0\mathbf{v}_0^T = \sqrt{\lambda_0}\bar{\mathbf{y}}\bar{\mathbf{x}}^T.$$

A.5.8 Neural tangent kernel

In this section, we review the neural tangent kernel (NTK). This object is useful because it directly describes the learning dynamics in output space $\hat{y}$ (Jacot et al., 2018; Roberts et al.) as we briefly demonstrate here. We then calculate the NTK for our specific architecture to yield **??** in the main text. The following makes use of Einstein summation convention.

For a vector-valued model $\hat{\mathbf{y}}(\mathbf{x}) \in \mathbb{R}^{N_{out}}$ parametrized by a parameter vector θ, the evaluation on sample $\mathbf{x}_i$ from training data at $\mathbf{x}_{i'}$ evolves as

$$\tau\frac{d}{dt}y_m(\mathbf{x}_i) = \sum_k \frac{dy_m(\mathbf{x}_i)}{d\theta^k}\frac{d\theta^k}{dt} \tag{11}$$

$$= -\eta\sum_k \frac{dy_m(\mathbf{x}_i)}{d\theta^k}\frac{d\mathcal{L}}{d\theta^k} \tag{12}$$

$$= -\eta\left[\sum_k \frac{dy_m(\mathbf{x}_i)}{d\theta^k}\frac{dy_{m'}(\mathbf{x}_{i'})}{d\theta^k}\right]\frac{d\mathcal{L}}{dy_{m'}}(\mathbf{x}_{i'}) \tag{13}$$

$$=: -\eta\,\mathsf{NTK}_{mm'}(\mathbf{x}_i,\,\mathbf{x}_{i'})\left(y_{m'}(\mathbf{x}_{i'}) - \hat{y}_{m'}(\mathbf{x}_{i'})\right), \tag{14}$$

where we used the chain rule and that the parameters update according to gradient descent with learning rate η, $\frac{d\theta^k}{dt} = -\eta \frac{d\mathcal{L}}{d\theta^k}$. The last line has defined the NTK. We set $\eta = 1$ in the main text for simplicity, as it does not change trajectory and thereby convergence in the case of gradient flow. In addition, we evaluated $\frac{d\mathcal{L}}{dy_{m'}}(\mathbf{x}_{i'})$ for the case of MSE loss $\mathcal{L}(\mathbf{x}_{i'}) = 1/2 \sum_{m'} \left(y_{m'}(\mathbf{x}_{i'}) - \hat{y}_{m'}(\mathbf{x}_{i'})\right)^2$. The last line of Eq. (11) reveals that the NTK acts as an effective learning rate, as noted by Roberts et al..

We here consider a two-layer linear architecture $\hat{Y}^i_m(\mathbf{X}) = W^2_{mk}\left(W^1_{kj}X^i_j + b^1_k\right) + b^2_m$ where we adopt Einstein summation convention over repeated indices. The parameters are $\theta^k \in \left\{\mathbf{W}^2, \mathbf{W}^2, \mathbf{b}^1, \mathbf{b}^2\right\}$. Herein, m indexes output features and i indexes data samples. The non-zero gradients are

$$
\begin{aligned}
\frac{d\hat{Y}^i_m}{dW^2_{mk}} &= W^1_{kj}X^i_j + b^1_k \\
\frac{d\hat{Y}^i_m}{db^2_m} &= 1_m \\
\frac{d\hat{Y}^i_m}{dW^1_{kj}} &= W^2_{mk}X^i_j \\
\frac{d\hat{Y}^i_m}{db^1_k} &= W^2_{mk}1_k.
\end{aligned}
$$

Inserting this into Eq. (11), we get

$$
\begin{aligned}
\mathsf{NTK}_{m_1m_2}(X^{i_1}_j, X^{i_2}_j) = I_{m_1m_2} & \left(X^{i_1}_{j'}W^1_{j'k}W^1_{kj''}X^{i_2}_{j''} + b^1_k b^1_k\right) \\
&+ 1_{m_1}1_{m_2} \\
&+ W^2_{m_1k}W^{2T}_{km_2}\, X^{i_1}_j X^{i_2}_j \\
&+ W^2_{m_1k}1_k 1_k W^2_{km_2}.
\end{aligned}
$$

or in matrix notation, collecting similar terms

$$
\begin{aligned}
\mathsf{NTK}(\mathbf{X}, \mathbf{X}) = \mathbf{I}_{N_{out}} & \otimes \mathbf{X}^T\mathbf{W}^{1T}\mathbf{W}^1\mathbf{X} + \mathbf{b}^{1T}\mathbf{b}^1 \\
&+ \mathbf{1}\mathbf{1}^T \otimes \underbrace{\mathbf{1}\mathbf{1}^T}_{\leftrightarrow \mathbf{b}^2} \\
&+ \mathbf{W}^2\mathbf{W}^{2T} \otimes \left(\mathbf{X}^T\mathbf{X} + \underbrace{\mathbf{1}\mathbf{1}^T}_{\leftrightarrow b^1}\right) \\
&\in \mathbb{R}^{N_{out}\times N_{out}} \otimes \mathbb{R}^{N\times N},
\end{aligned}
$$

where the left hand side operator in the tensor product $\otimes$ is acting in output space m_1m_2, whereas the right hand side operator acts in pattern space i_1i_2. The notation $\leftrightarrow \mathbf{b}$ indicates that a term is due to the bias term. To illustrate this, the NTK acts on the set of labels $\mathbf{Y} \in \mathbb{R}^{N_{in}\times N}$ as follows:

$$
\left(\mathsf{NTK}(\mathbf{X}, \mathbf{X})\, \mathbf{Y}\right)^m_i = \sum_{m'}^{N_{out}} \sum_{i'}^{N} \mathsf{NTK}(X_i, X_{i'})^{mm'} Y^{m'}_{i'}. \tag{15}
$$

For simplicity, we approximate $\mathbf{W}^2(0)\mathbf{W}^{2T}(0) = \sigma^2_{\mathbf{W}} I_{N_{out}}$ and $\mathbf{W}^{1T}(0)\mathbf{W}^1(0) = \sigma^2_{\mathbf{W}} I_{N_{in}}$, which approximately holds for initialization

$$
\mathbf{W}^1(0) \sim \mathcal{N}\left(0,\, \sigma^2_{\mathbf{W}}/N_{hid}\right),\ \mathbf{W}^2(0) \sim \mathcal{N}\left(0,\, \sigma^2_{\mathbf{W}}/N_{hid}\right),\ \mathbf{b}^1 = 0,\ \mathbf{b}^2 = 0
$$

where N_{hid} is the size of the hidden layer and both N_{in} and N_{hid} are large. This leaves

$$
\mathsf{NTK}(\mathbf{X}, \mathbf{X}) = \sigma^2_{\mathbf{W}}\mathbf{I}_{N_{out}} \otimes \left(2\mathbf{X}^T\mathbf{X} + \underbrace{\mathbf{1}\mathbf{1}^T}_{\leftrightarrow b^1}\right) + \mathbf{1}\mathbf{1}^T \otimes \underbrace{\mathbf{1}\mathbf{1}^T}_{\leftrightarrow \mathbf{b}^2}.
$$

A.6 SHALLOW NETWORK OCS LEARNING

In this brief section we show the OCS signatures of shallow networks with bias terms. The result is displayed in Fig. 11. We see similar behavioural signatures to deep linear networks. However, the tendency to the OCS is more transient.

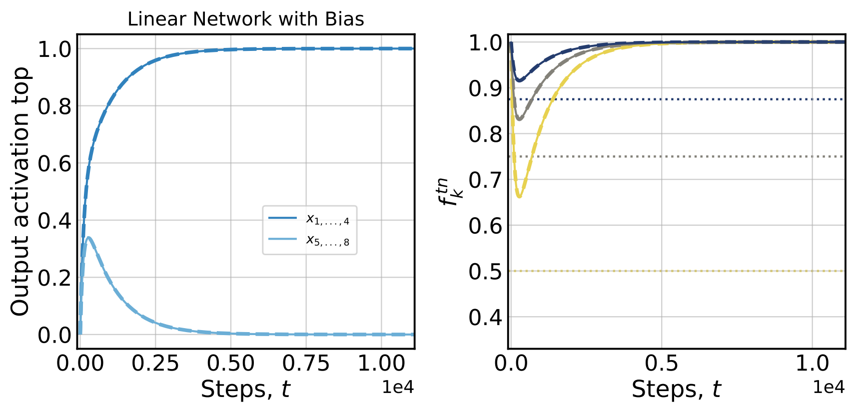


Figure 11: Early learning in shallow networks with bias terms approaches the OCS.

A.7 TRUE NEGATIVE RATES IN LINEAR NETWORKS WITHOUT BIAS TERMS

In this short section we provide a supplemental figure relevant for our results in Section 5: We train deep and shallow linear networks *without* bias terms. The learning setting and computation of metrics are equivalent to results in Fig. 4. We display the result in Fig. 12. While networks learn the task, early, response biases are fully absent in these models.

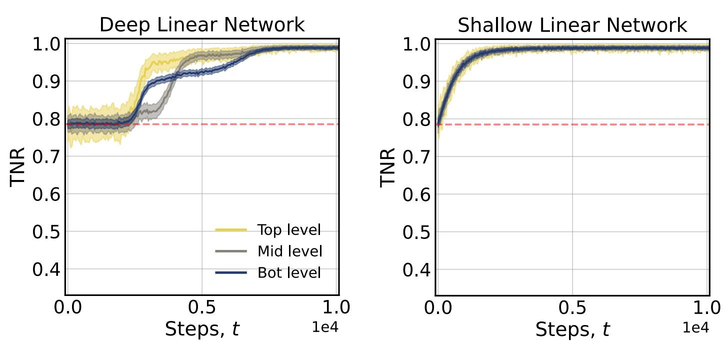


Figure 12: True negative rates for linear networks *without* bias terms. We do not see characteristic response patterns observed in Fig. 4.

A.8 CNN DATASETS AND HYPERPARAMETERS

Datasets used. We used and adapted different image datasets for our experiments with CNNs. While the main text focused on results obtained with a variant of MNIST we report further experiments we conducted to highlight the generality of early OCS learning.

1. **Hierarchical MNIST.** We used the default ten digit classes provided by MNIST. We then sampled 8 classes randomly and replaced the default one-hot labels corresponding to each class i with the hierarchical, “three-hot” labels y_i as seen in Fig. 1. E.g., all images corresponding to MNIST digit “1” might be assigned some random “three-hot” output vector y_i.

2. **Hierarchical CIFAR-10.** We applied the same procedure and randomly sampled eight classes from CIFAR-10 and replaced one-hot labels as for the hierarchical MNIST.
3. **Imbalanced-binary-MNIST.** While not described in the main text we also report results in a setting with standard one-hot target vectors. We randomly sample two MNIST classes for training. To assess the impact of the OCS in early learning we introduced class imbalanced by oversampling one of the two classes by a factor of two.
4. **Standard CelebA.** We perform experiments on CelebA's face attribute detection task. The task offers a natural testbed for early learning of the OCS as face attribute target labels form a non-uniform distribution as seen in Fig. 15, bottom. We also normalised images in the dataset before training.

Model details. We trained a custom CNN with 3 convolutional layers (layer 1: 32 filters of size 5×5; layer 2: 64 filters of size 3×3; layer 3: 96 filters of size 3×3), followed by 2 fully connected layers of sizes 512 and 256. Activation functions for all layers were chosen as ReLUs. The final layer of the model did not contain an activation function when training with squared error loss. In experiments with the class imbalanced-binary-MNIST and cross-entropy loss the final layer contained a softmax function as non-linearity. For experiments on CelebA the final layer contained sigmoid activation functions and we trained with a binary cross-entropy loss over all 40 labels.

Training details. For our results on hierarchical MNIST we train models with minibatch SGD with a batch size of 16 and with a relatively small step size of 1e-4 to examine the early learning phase. For all experiments we used Xavier uniform initialisation (Glorot and Bengio, 2010). Whenever we use bias terms in the model we initialize these as 0 in line with common practice. For our main experiments we train models using a simple squared error loss function. However, to demonstrate generality we repeat experiments for the case of class imbalance using a cross-entropy loss and binary cross-entropy in the case of CelebA. All experiments are repeated 10 times with different random seeds with the exception of CelebA where we used 5 different random seeds, we provide standard errors in all figures (shaded regions). For experiments on the hierarchical CIFAR-10, the class imbalanced MNIST, and CelebA we kept all parameters as above but we increase step size to 1e-3. We trained CNN models on an internal cluster on a single RTX 5000 GPU. Runs took less than one hour to complete.

A.9 ADDITIONAL EXPERIMENTAL RESULTS

To understand the generality of OCS learning we plot the results of experiments examining early learning of the OCS in these models. We mostly restrict ourselves to plots as seen in Fig. 5 as we deem these figures most instructive.

Hierarchical CIFAR-10. We train on a hierarchical version of CIFAR-10. Where we randomly sample 8 classes from MNIST and replace target labels by hierarchical vectors as in Section 4. We find the key signatures of early OCS learning: We find early indifference, the reversion of performance metrics to the OCS, and a small initial distance of average response the to the OCS solution. The results mirror behaviour on the hierarchical MNIST shown in Fig. 3 and Fig. 5.

Class-imbalance MNIST. We train on an imbalanced MNIST task as described in Appendix A.8. We plot the results for training with squared error and cross-entropy loss functions in Fig. 14. Both settings show reversion to the OCS. Note that average model outputs in the case of the cross-entropy loss start relatively close to outputs expected under the OCS. Despite this proximity the model is still driven towards the OCS solution. The results on this imbalanced case highlight potential fairness implications. Given that network have been found to revert to the OCS when generalising (Kang et al., 2024), early learning in the OCS setting can transiently, but significantly disadvantage minority classes. We further highlight this point in a second solvable case of linear networks with bias terms in Appendix A.10.

Standard CelebA. We show distance from the OCS for the CelebA face attribute detection task in Fig. 15, top. CelebA provides a useful test for our hypothesis as attribute labels display natural imbalances. We highlight the strong non-uniformity of the majority attribute labels in Fig. 15, bottom. We again train networks in two variants: one with squared-error loss and one with binary cross-entropy loss applied over all 40 face attributes. With both loss functions network responses are driven towards the OCS in early learning. This case further highlights the generality of early OCS

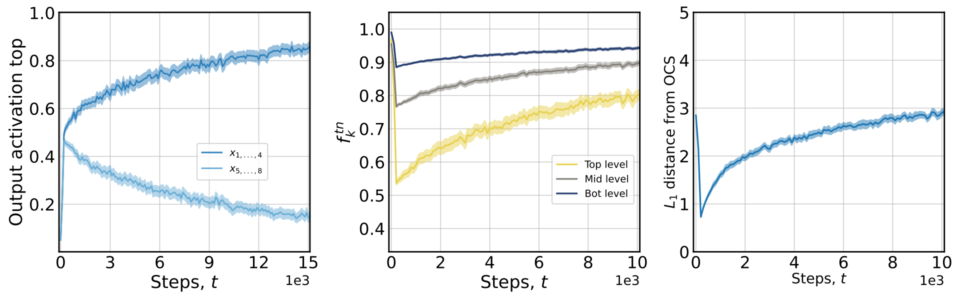


Figure 13: Early OCS learning CNNs trained on hierarchical CIFAR-10. *left:* Network outputs for a single output unit in response to all inputs $\mathbf{x}_i$. *Centre:* Performance metrics f^{tn} (Appendix A.4). *Right:* Mean distance of network responses from OCS. Averages taken over every 10 batches for plotting.

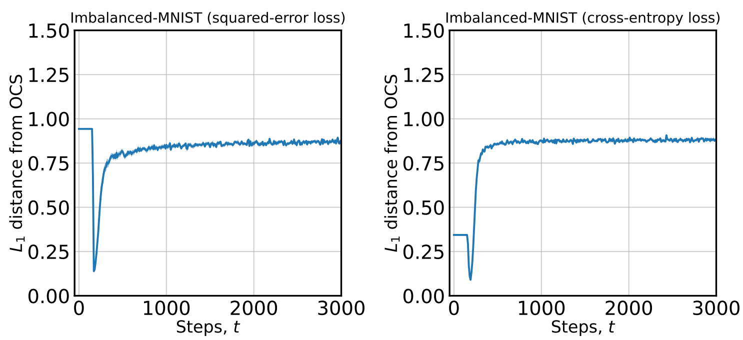


Figure 14: Mean distance of network responses from OCS in CNNs trained on the imbalanced MNIST task. Averages taken over each batch.

learning. OCS learning might be especially undesirable in this setting for fairness reasons as the model will be overly liberal in the prediction of the most common face attributes.

A.10 Linear networks under class imbalance

In this section, we describe a second case of a solvable linear network with bias terms. Our dataset consists of two examples where one example appears twice as frequently. We show the data used on the right side of Fig. 16. The minority class has two identifying labels, while this construction appears artificial, it allows for the application of Proposition 1 and solutions to learning dynamics from Section 2 apply.

The case is of particular practical relevance as it illustrates the impact of early OCS learning under class imbalance, a common problem in machine learning where datasets are often naturally imbalanced (Feldman, 2020; Van Horn and Perona, 2017). In practice, these settings are often addressed through oversampling of minority classes (Haibo He and Garcia, 2009; Huang et al., 2016). Empirical work by Ye et al. (2021) documented that neural networks initially fail to learn information about the minority class while classifying most minority examples as belonging to the majority class. Subsequent theoretical work by Francazi et al. (2023) demonstrated that the phenomenon is caused by competition between the optimisation of different classes.

Our work adds to this literature by providing dynamics in a case of gradient-based learning under class imbalance learning that is exactly solvable. Our exact solutions highlight the potential role of early OCS learning in the initial failure to learn about minority classes. The OCS solution substantially biases early predictions towards the majority class as seen in Fig. 16, centre. The results also can be understood as solvable analogous to early reversion to the OCS seen in the Imbalanced-binary-MNIST setting in Fig. 14. The results highlight the potential fairness implications of early OCS learning as the learning phase systematically biases the model against the minority classes.

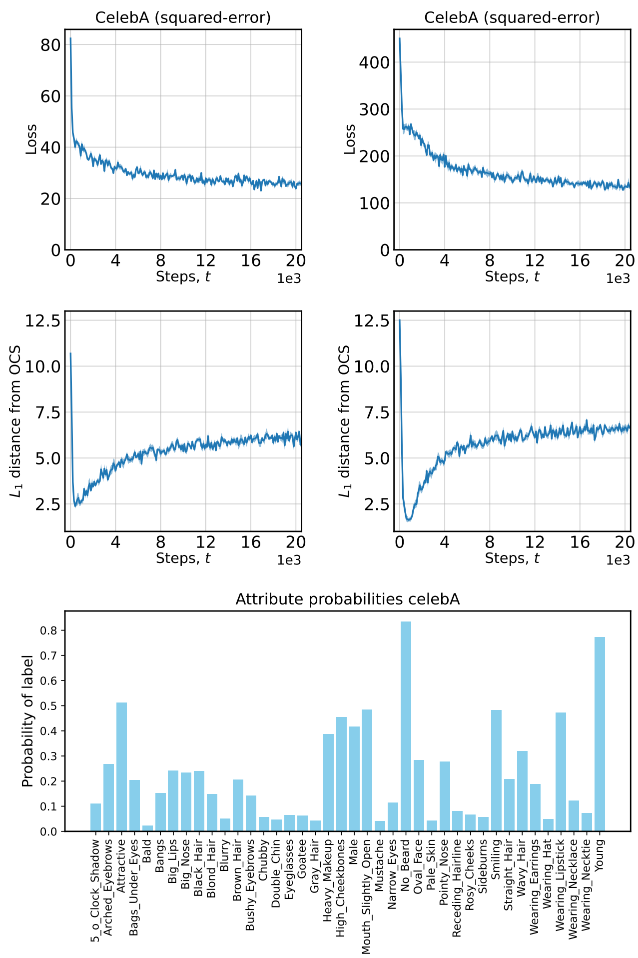


Figure 15: *Top:* Mean distance of network responses from the OCS in CNNs trained on the CelebA face attribute prediction task. *Bottom:* Marginal probabilities of CelebA face attributes.

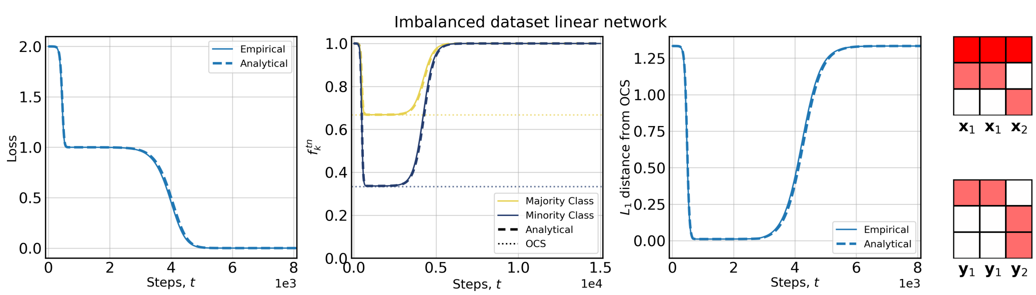


Figure 16: Early learning of the OCS in linear networks under class imbalance.

A.11 OCS-LEARNING IN LINEAR NETWORKS WITH INPUT CORRELATIONS.

In this section, we demonstrate how input correlations can drive OCS learning in the absence of bias terms in linear networks. Specifically we highlight how OCS learning can emerge if $\mathbf{1}_N$ is an eigenvector of the data similarity matrix $\mathbf{X}^T\mathbf{X}$. Note that the network contains no input correlations. In the bottom row of the Fig. 17, we can see that the first SVD mode $\mathbf{u}_1\mathbf{v}_1^T$ is indeed exactly equivalent to the OCS mode, i.e. $\bar{\mathbf{y}}\bar{\mathbf{x}}^T$. The right panel highlights how the network is driven towards the OCS up until the time-point when the second effective singular value $a_2(t)$ (which is quite close in time) is learned.

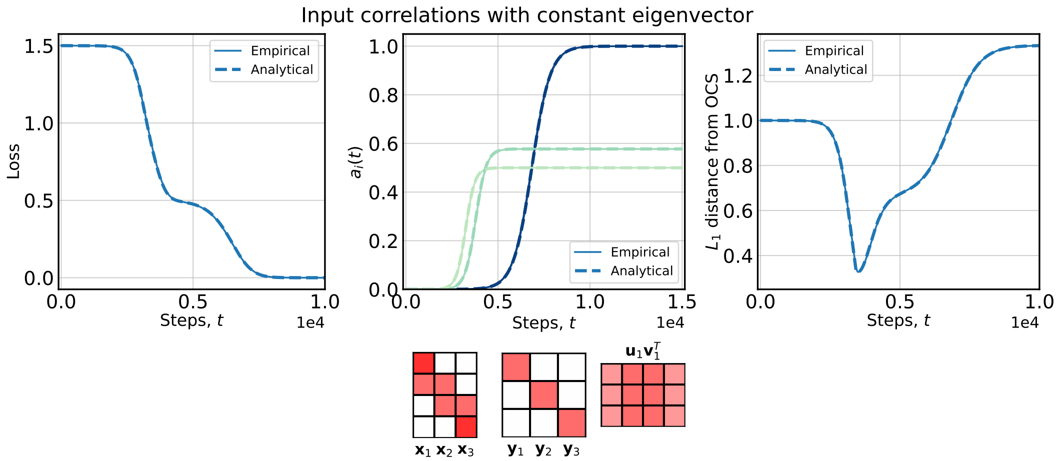


Figure 17: Early learning of the OCS in linear networks with input correlations but in the absence of bias terms.

A.12 Relation to imbalanced multi-label learning

Given the hierarchical structure of labels used in the majority of our experiments we also see some general connections of our work to problems in the domain of imbalanced multi-label learning. Multi-label learning deals with learning problems in which a single input example is associated with multiple output labels simultaneously. In these settings class imbalance is a key challenge that frequently hinders good performance of models (Liu et al., 2020; Charte et al., 2015; Pham et al., 2021; Liu et al., 2022). Similar to standard classification problems model biases are frequently addressed through adjustments to the models loss function via selective reweighing (Cui et al., 2019) or through sampling based methods which selectively over- or under-sample particular labels (Charte et al., 2015) or via both methods (Pham et al., 2021). Our results on the hierarchical learning task and on the problem of class imbalanced learning in Appendix A.10 might hint at OCS learning as a potential contributor to problems observed in multi-label learning as the imbalanced distribution of output labels might drive learning to undesirable solutions.