Not all solutions are created equal: An analytical dissociation of functional and representational similarity in deep linear neural networks

Lukas Braun [1] **Erin Grant** [2] **Andrew M. Saxe** [2]

Abstract

A foundational principle of connectionism is that perception, action, and cognition emerge from parallel computations among simple, interconnected units that generate and rely on neural representations. Accordingly, researchers employ multivariate pattern analysis to decode and compare the neural codes of artificial and biological networks, aiming to uncover their functions. However, there is limited analytical understanding of how a network's representation and function relate, despite this being essential to any quantitative notion of underlying function or functional similarity. We address this question using fully analysable two-layer linear networks and numerical simulations in nonlinear networks. We find that function and representation are dissociated, allowing representational similarity without functional similarity and vice versa. Further, we show that neither robustness to input noise nor the level of generalization error constrain representations to the task. In contrast, networks robust to parameter noise have limited representational flexibility and must employ task-specific representations. Our findings suggest that representational alignment reflects computational advantages beyond functional alignment alone, with significant implications for interpreting and comparing the representations of connectionist systems.

1. Introduction

The *parallel distributed processing* hypothesis posits that function in artificial and biological networks emerges from interactions among simple interconnected units that compute with distributed representations (Rumelhart et al. 1986). Accordingly, one might aim to identify function from networks observables such as connectivity weights and neural activity patterns; however, this is often complicated by the inherent complexity and partial observability of these systems. In particular, the structure of artificial and biological networks is often *non-identifiable* in the sense that networks can be structurally distinct, yet implement the same input-output mapping. For example, biophysical neuron models can exhibit nearly identical functions at both the neuron (Goldman et al. 2001) and network levels (Prinz et al. 2004) despite considerable variation in their architecture (reviewed in Marder and Goaillard 2006; Albantakis et al. 2024). Similarly, artificial neural networks (ANNs) are almost always non-identifiable due to simple symmetries, such as permutation-invariance of neurons (Sussmann 1992; Albertini and Sontag 1993), scale-invariance of activation functions (Neyshabur et al. 2015a), alongside more complex symmetries arising from feature composition across layers and from finite training data (Refinetti et al. 2021; Arous et al. 2022). As modern networks are deep and heavily overparameterised (Zhang et al. 2021), they are inherently non-identifiable, with many parametrisations yielding the same input-output behaviour. Determining when parametrisations become identifiable and understanding the consequences of non-identifiability remain open problems (Roeder et al. 2021; Entezari et al. 2022; Vlačić and Bölcskei 2022; Ghosh et al. 2022; Wang and Jordan 2021; Godfrey et al. 2022; Martinelli et al. 2023; Bona-Pellissier et al. 2021; Kori et al. 2024; Marconato et al. 2024).

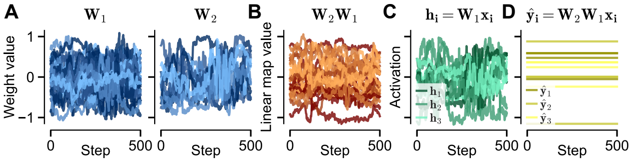

Figure 1: Random walk. **(A)** A random walk on the solution manifold of a two-layer linear network reveals that input and readout weights can change continuously, inducing changes in the **(B)** network parametrisation and thus the **(C)** hidden-layer representations, while preserving the **(D)** network output.

Even deep linear networks exhibit both trivial and non-trivial symmetries, making their parametrisation non-identifiable. While any deep linear network can be re-parametrised as

[1] Department of Experimental Psychology, University of Oxford, Oxford, UK [2] Gatsby Unit & Sainsbury Wellcome Centre, University College London, London, UK. Correspondence to: Lukas Braun <lukas.braun@psy.ox.ac.uk>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

a single linear transformation (Laurent and Brecht 2018), it does so through multistage computations that give rise to hidden-layer representations. Moreover, the optimisation landscape of a deep linear network is non-convex and contains a high-dimensional solution manifold (Figure 1) whose shape is determined by the statistics of training data and the network architecture (Baldi and Hornik 1989; Saxe et al. 2014; Arora et al. 2019), making it a useful surrogate for studying representation learning (Saxe et al. 2019; Braun et al. 2022; Dominé et al. 2024). Here, we leverage the analytical tractability of deep linear networks to study functionally equivalent parametrisations at global minimum error. Crucially, these solutions employ different internal representations, which has significant computational consequences, most notably in their affordances for linear decoding, representational similarity analysis (Section 4), and their sensitivity to noise (Section 5).

We now detail our **contributions:**

- We derive exact parametric equations characterising the complete and distinct subregions of the solution manifold in two-layer linear networks.
- We demonstrate that, although all subregions allow flexible neural representations, some inherently lead to identifiable task-specific representational similarities, while others result in non-identifiable, task-agnostic representational similarities.
- We establish that in contrast to task-specific solutions, task-agnostic solutions are non-identifiable and non-comparable.
- We analytically show that input noise and generalisation error do not constrain representations to task-specific regions, whereas parameter noise does.
- We validate our analytical findings through numerical simulations, demonstrating that these computational principles persist in non-linear neural networks.

All simulations are detailed in Appendix A, and a code repository reproducing all figures is available on GitHub at lukas-braun/dissociating-similarity.

2. Setting and preliminaries

We consider a two-layer linear network (Figure 1A),

$$\hat{\mathbf{y}}_n = \mathbf{W}_2 \mathbf{W}_1 \mathbf{x}_n, \tag{1}$$

trained to minimise the mean-squared error

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2P} \sum_{n=1}^{P} ||\hat{\mathbf{y}}_n - \mathbf{y}_n||_2^2 \tag{2}$$

over a dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{P}$, with inputs $\mathbf{x}_n \in \mathbb{R}^{N_i}$ and corresponding targets $\mathbf{y}_n \in \mathbb{R}^{N_o}$. The input weights $\mathbf{W}_1 \in \mathbb{R}^{N_h \times N_i}$ project inputs to hidden-layer neural representation $\mathbf{h}_n = \mathbf{W}_1 \mathbf{x}_n \in \mathbb{R}^{N_h}$, which are projected to outputs via the readout weights $\mathbf{W}_2 \in \mathbb{R}^{N_o \times N_h}$. We denote by $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_P]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_P]$, and $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_P]$ the matrices that contain all inputs, targets and hidden-layer representations, respectively. The network's representational similarity matrix (RSM) is then defined by

$$\text{RSM} = \mathbf{X}^T \mathbf{W}_1^T \mathbf{W}_1 \mathbf{X} = \mathbf{H}^T \mathbf{H}, \tag{3}$$

capturing pairwise similarities between inputs in the hidden representational space. Laurent and Brecht (2018) showed that, under the following assumptions:

Assumption 2.1. The loss function is convex and differentiable, *e.g.*, the mean-squared error loss.

Assumption 2.2. The network is not bottlenecked, *i.e.*, $\min(N_i, N_o) \leq N_h$

all local minima of a deep linear network are global and equivalent to the solution of the corresponding single-layer linear regression problem. Notably, this result holds without assumptions on the structure of the training data. In our setting, this permits the following definition (see Appendix C):

Definition 2.3. Under Assumptions 2.1 and 2.2, any pair of network weights satisfying

$$\mathbf{W}_2 \mathbf{W}_1 \mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx} \tag{4}$$

is a globally optimal general linear solution (GLS), where

$$\mathbf{\Sigma}_{xx} = \frac{1}{P} \sum_{n=1}^{P} \mathbf{x}_n \mathbf{x}_n^T \quad \text{and} \quad \mathbf{\Sigma}_{yx} = \frac{1}{P} \sum_{n=1}^{P} \mathbf{y}_n \mathbf{x}_n^T \tag{5}$$

denote the input and input-output covariance matrices.

Note that the absence of suboptimal minima does not preclude the existence of other critical points, nor does it guarantee convergence of gradient-based algorithms. To distinguish general weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ from those that satisfy Equation (4), we denote the latter as optimal weights $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. Further, we denote the compact singular value decomposition (cSVD), as defined in Appendix B.1, of the inputs and least-squares solution as

$$\text{cSVD}(\mathbf{X}) = \mathbf{A}\mathbf{B}\mathbf{C}^T, \tag{6}$$

and

$$\text{cSVD}(\mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+) = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{7}$$

In the following, we analytically study the full set of global solutions, the so called solution manifold, and partition it into subregions with distinct representational and computational properties. Crucially, our analysis holds without assumptions on the structure of the training data and irrespective of how a solution is obtained, and thus does not depend on any particular learning or optimisation algorithm. Figure 2A visualises the entire solution manifold and its subregions for a simple example, providing some intuition for the formal definitions and theorems developed next.

3. Partitioning the solution manifold of two-layer linear networks

Two-layer linear networks architectures are typically highly overparametrised, admitting many combinations of input and output weights that achieve the global optimum for a given task. Formally, the set of all such network weights defines the solution manifold,

$$\mathcal{M} = \left\{ \mathbf{\Omega}_2 \mathbf{\Omega}_1 : \mathbf{\Omega}_2 \mathbf{\Omega}_1 \mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx} \right\} . \tag{8}$$

We note that useful intuition about the manifold's structure can be gained by viewing it as the set of weight configurations related by invertible linear transformations. For any invertible matrix $\mathbf{Q} \in \mathcal{R}^{N_h \times N_h}$, the weight pair

$$\mathbf{\Omega}_2 \to \mathbf{\Omega}_2 \mathbf{Q}^{-1} \quad \text{and} \quad \mathbf{\Omega}_1 \to \mathbf{Q} \mathbf{\Omega}_1 \tag{9}$$

implements the same input-output map and thus lies on the same manifold (Baldi and Hornik 1989; Saxe et al. 2014). In the context of a neural network architecture, these $\mathbf{Q}$-transformations redistribute how information is processed across layers, for example, by rotating and scaling intermediate representations. However, since they preserve rank, they only fully characterise the solution manifold when the input and output dimensions are identical and the task has full rank, conditions that may not be met in real-world scenarios. To refine this view, we partition the input space into three subspaces: *Observed and relevant*, *observed but irrelevant*, and *unobserved* null directions, with corresponding projections $\mathbf{P}_\mathrm{r} = \mathbf{V}\mathbf{V}^T$, $\mathbf{P}_\mathrm{i} = \mathbf{A}\mathbf{A}^T - \mathbf{V}\mathbf{V}^T$, and $\mathbf{P}_\mathrm{u} = \mathbf{I} - \mathbf{A}\mathbf{A}^T$. The distinction between relevant and irrelevant directions arises because the input space can exceed the intrinsic dimensionality of the solution manifold (but not vice versa). Specifically,

$$r = \operatorname{rank}(\mathbf{\Sigma}_{yx}) \leq \min(\operatorname{rank}(\mathbf{X}), \operatorname{rank}(\mathbf{Y})), \tag{10}$$

so when the input rank exceeds the target rank, some input directions are irrelevant to solving the task. Likewise, the hidden space can be partitioned into subspaces corresponding to the hidden-layer representations of relevant, and irrelevant inputs, and all remaining unoccupied null directions. Importantly, the hidden representations of relevant and irrelevant inputs may overlap, which necessitates compensation and introduces structural constraints on the form of valid solutions. The following parametrized equation fully encapsulates this intricate structure of the solution manifold:

Theorem 3.1. *Any GLS satisfies*

$$\begin{aligned} \mathbf{\Omega}_1 &= \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{\Gamma}_1 \mathbf{P}_i + \mathbf{\Gamma}_2 \mathbf{P}_u \quad \textit{and} \\ \mathbf{\Omega}_2 &= \mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \mathbf{\Psi} + \mathbf{\Gamma}_3(\mathbf{I} - \mathbf{H}\mathbf{H}^+), \end{aligned} \tag{11}$$

where $\mathbf{Q} \in \mathcal{R}^{N_h \times r}$ *is an arbitrary full-column-rank matrix,* $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2 \in \mathcal{R}^{N_h \times N_i}$ *are arbitrary matrices subject to the constraint* $\operatorname{rank}(\mathbf{Q}\mathbf{Q}^+\mathbf{\Gamma}_1\mathbf{P}_i) \leq \operatorname{rank}((\mathbf{I} - \mathbf{Q}\mathbf{Q}^+)\mathbf{\Gamma}_1\mathbf{P}_i)$*,* $\mathbf{\Psi} = -\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+\mathbf{\Gamma}_1\mathbf{P}_i \left[(\mathbf{I} - \mathbf{Q}\mathbf{Q}^+)\mathbf{\Gamma}_1\mathbf{P}_i\right]^+$*, and* $\mathbf{\Gamma}_3 \in \mathcal{R}^{N_o \times N_h}$ *is an arbitrary matrix.*

The first terms of $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ implement the core input-output mapping; $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ project from task-irrelevant and unobserved input directions; $\mathbf{\Gamma}_3$ projects from the unoccupied hidden space; $\mathbf{\Psi}$ cancels interference from irrelevant inputs that are projected into the core; and the rank constraint ensures that such a correction exists. See Appendix C for a detailed proof and Figure 2B for a visualisation.

Next, we partition the solution manifold into distinct regions and subsequently analyse their respective representational and computational properties. For proofs of Theorems 3.3, 3.5 and 3.7 refer to Appendix D, and to Figure 2C-E for visualisations.

Definition 3.2. Any GLS that minimises the norm of the network function,

$$\underset{\mathbf{W}_1, \mathbf{W}_2}{\operatorname{argmin}} \, ||\mathbf{W}_2 \mathbf{W}_1||_F^2 \quad \text{s.t.} \quad \mathbf{W}_2 \mathbf{W}_1 \mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx} \tag{12}$$

is a least-squares solution (LSS).

Theorem 3.3. *All LSS satisfy* $\mathbf{\Omega}_2 \mathbf{\Omega}_1 = \mathbf{\Sigma}_{yx} \left(\mathbf{\Sigma}_{xx}\right)^+$ *and are exactly parametrised by*

$$\begin{aligned} \mathbf{\Omega}_1 &= \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{\Gamma}_1 \mathbf{P}_i + \mathbf{\Gamma}_2 \mathbf{P}_u \quad \textit{and} \\ \mathbf{\Omega}_2 &= \mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \mathbf{\Psi} + \mathbf{\Phi} + \mathbf{\Gamma}_3(\mathbf{I} - \mathbf{\Omega}_1 \mathbf{\Omega}_1^+), \end{aligned} \tag{13}$$

subject to the definitions and constraints in Theorem 3.1, and the additional constraint that $\operatorname{rank}(\mathbf{H}\mathbf{H}^+\mathbf{\Omega}_1\mathbf{P}_u) \leq \operatorname{rank}((\mathbf{I} - \mathbf{H}\mathbf{H}^+)\mathbf{\Omega}_1\mathbf{P}_u)$*, and where* $\mathbf{\Phi} = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \mathbf{\Psi})\mathbf{\Omega}_1\mathbf{P}_u \left[(\mathbf{I} - \mathbf{H}\mathbf{H}^+)\mathbf{\Omega}_1\mathbf{P}_u\right]^+$.

Here, $\mathbf{\Phi}$ cancels interference from unobserved inputs projected into the occupied hidden space, with the rank constraint ensuring a correction exists.

Definition 3.4. Any GLS for which the norm of the hidden-layer representations and readout weights is minimised

$$\begin{aligned} &\underset{\mathbf{W}_1, \mathbf{W}_2}{\operatorname{argmin}} \, ||\mathbf{W}_1 \mathbf{X}||_F^2 + ||\mathbf{W}_2||_F^2 \\ &\qquad\qquad \text{s.t.} \; \mathbf{W}_2 \mathbf{W}_1 \mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx} \; , \end{aligned} \tag{14}$$

is a minimum representation-norm solution (MRNS).

Theorem 3.5. *All MRNS are parametrised by*

$$\mathbf{\Omega}_2 = \mathbf{M}\sqrt{\mathbf{N}}\mathbf{R}^T \;\; \textit{and} \;\; \mathbf{\Omega}_1 = \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+ + \mathbf{\Gamma}_2\mathbf{P}_u, \tag{15}$$

where $\mathbf{R} \in \mathcal{R}^{N_h \times r}$ *is an arbitrary (semi-)orthonormal matrix, and*

$$\operatorname{cSVD}(\mathbf{Y}\mathbf{C}\mathbf{C}^T) = \mathbf{M}\mathbf{N}\mathbf{O}^T. \tag{16}$$

Definition 3.6. Any GLS for which the sum of the norm of the weight matrices is minimised

$$\underset{\mathbf{W}_1, \mathbf{W}_2}{\operatorname{argmin}} \, ||\mathbf{W}_1||_F^2 + ||\mathbf{W}_2||_F^2 \; \text{s.t.} \; \mathbf{W}_2 \mathbf{W}_1 \mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx} \tag{17}$$

is a minimum weight-norm solution (MWNS).

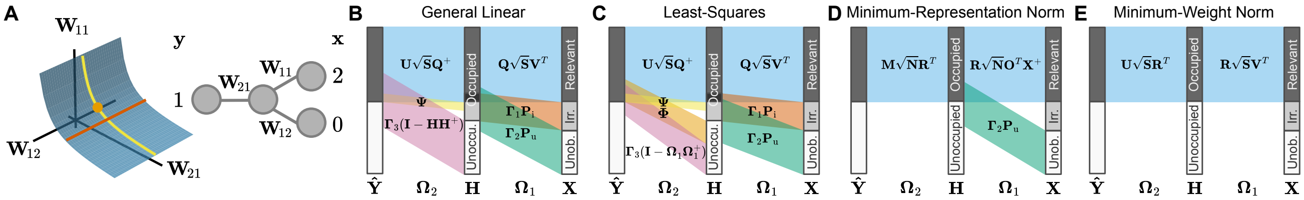
A
$\mathbf{W}_{11}$
$\mathbf{W}_{12}$
$\mathbf{W}_{21}$
y
x
1
2
0
$\mathbf{W}_{21}$
$\mathbf{W}_{11}$
$\mathbf{W}_{12}$
B
General Linear
$\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+$
$\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T$
$\mathbf{\Psi}$
$\mathbf{\Gamma}_3(\mathbf{I}-\mathbf{H}\mathbf{H}^+)$
$\mathbf{\Gamma}_1\mathbf{P}_\mathrm{i}$
$\mathbf{\Gamma}_2\mathbf{P}_\mathrm{u}$
Occupied
Unoccu.
Relevant
Irr.
Unob.
$\hat{\mathbf{Y}}$
$\mathbf{\Omega}_2$
$\mathbf{H}$
$\mathbf{\Omega}_1$
$\mathbf{X}$
C
Least-Squares
$\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+$
$\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T$
$\mathbf{\Psi}$
$\mathbf{\Phi}$
$\mathbf{\Gamma}_3(\mathbf{I}-\mathbf{\Omega}_1\mathbf{\Omega}_1^+)$
$\mathbf{\Gamma}_1\mathbf{P}_\mathrm{i}$
$\mathbf{\Gamma}_2\mathbf{P}_\mathrm{u}$
Occupied
Unoccu.
Relevant
Irr.
Unob.
$\hat{\mathbf{Y}}$
$\mathbf{\Omega}_2$
$\mathbf{H}$
$\mathbf{\Omega}_1$
$\mathbf{X}$
D
Minimum-Representation Norm
$\mathbf{M}\sqrt{\mathbf{N}}\mathbf{R}^T$
$\mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+$
$\mathbf{\Gamma}_2\mathbf{P}_\mathrm{u}$
Occupied
Unoccupied
Relevant
Irr.
Unob.
$\hat{\mathbf{Y}}$
$\mathbf{\Omega}_2$
$\mathbf{H}$
$\mathbf{\Omega}_1$
$\mathbf{X}$
E
Minimum-Weight Norm
$\mathbf{U}\sqrt{\mathbf{S}}\mathbf{R}^T$
$\mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T$
Occupied
Unoccupied
Relevant
Irr.
Unob.
$\hat{\mathbf{Y}}$
$\mathbf{\Omega}_2$
$\mathbf{H}$
$\mathbf{\Omega}_1$
$\mathbf{X}$

Figure 2: Solution manifold. **(A)** Schematic of solution manifold (left) for a two-layer linear network trained on a single training pair (right). The GLS (blue plane) reflects that weight $\mathbf{W}_{12}$ lies in the input null space and is unconstrained, while $\mathbf{W}_{11}$ and $\mathbf{W}_{21}$ are coupled, an increase in one requires a decrease in the other. Constrained LSS (yellow), MRNS (red), MWNS (orange) are highlighted subregions of the manifold. **(B)** Schematic of the parametrisation of the GLS, showing how components of $\mathbf{\Omega}_1$ map relevant, irrelevant, and unobserved input directions to the hidden space, and how components of $\mathbf{\Omega}_2$ map from unoccupied and occupied hidden directions to the output. Projections from irrelevant inputs can interfere with the core (blue), creating overlap (black) between the relevant (dark grey) and irrelevant (light grey) hidden space , which is cancelled by $\mathbf{\Psi}$. **(C)** As in **(B)** , but for LSS. Projections from unobserved input directions into the occupied hidden space are cancelled by $\mathbf{\Phi}$. **(D)** and **(E)** are as in **(B)** , but show MRNS and MWNS respectively. The additional constraints remove projections and further restrict the core.

Theorem 3.7. *All MWNS are parametrised by*

$$\mathbf{\Omega}_2 = \mathbf{U}\sqrt{\mathbf{S}}\mathbf{R}^T \quad \textit{and} \quad \mathbf{\Omega}_1 = \mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T. \tag{18}$$

We note, that the relation between MWNS and the singular value decomposition of the least-squares solution has been previously derived under strong assumptions, namely that $\mathbf{\Sigma}_{xx} = \mathbf{I}$, $N_i = N_o$ and that $\mathbf{\Sigma}_{yx}$ has full rank (Saxe et al. 2019, see appendix S14-S15). Further, we note that

Corollary 3.8. *MRNS and MWNS are identical if inputs are whitened, i.e.,* $\mathbf{\Sigma}_{xx} = \mathbf{I}$.

A key difference between the four solution types lies in how they constrain the image and kernel of the weight matrices, which map between input, hidden, and output spaces. GLS impose minimal constraints. LSS restrict projections to and from null spaces in the input and hidden layers. MRNS further reduce freedom in the null space, eliminate irrelevant projections, and constrain the core solution itself. MWNS, the most restrictive class, eliminate all irrelevant and null-space projections, and enforce balance in the core by requiring equal contributions from the input and output weights. While this perspective clarifies how different subregions of the solution manifold constrain the structure of the weight matrices, it does not address a key question: how these solutions differ in their hidden-layer representations.

3.1. Hidden-layer representations

Understanding hidden-layer representations begins with identifying the degrees of freedom in the input weights, which govern how inputs are mapped into hidden space. Intuitively, input weights are constrained only by the need to preserve sufficient task-relevant information for the output weights to solve the task. In this section, we go beyond this intuition by leveraging the exact parametrisations of $\mathbf{\Omega}_1$ to precisely characterise the degrees of freedom in hidden-layer representations. Proofs for Corollaries 3.9, 3.11 and 3.12 are in Appendix E. We begin by noting that GLS and LSS differ only in how they handle projections from unobserved input and unoccupied hidden directions and thus implement the same input-output map on the training data. As a result, they exhibit identical degrees of freedom in their hidden-layer representations

$$\mathbf{H} = \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T\mathbf{X} + \mathbf{\Gamma}_1\mathbf{P}_\mathrm{i}\mathbf{X}. \tag{19}$$

Since $\mathbf{Q}$ can be any full-column-rank matrix and $\mathbf{\Gamma}_1$ is free up to a rank constraint, GLS and LSS support nearly arbitrary hidden-layer representations. To illustrate this, we consider a semantic learning task linking items to positions within a hierarchical structure (Figure 3A,B). For example, we can select a point on the solution manifold where the hidden-layer representations of the items form the shape of an elephant (Figure 3C).

Corollary 3.9. *GLS and LSS permit any RSM of the form*

$$\begin{aligned} \mathrm{RSM} = \mathbf{X}^T(&\mathbf{V}\sqrt{\mathbf{S}}\mathbf{Q}^T\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{V}\sqrt{\mathbf{S}}\mathbf{Q}^T\mathbf{\Gamma}_1\mathbf{P}_i \\ &+ \mathbf{P}_i^T\mathbf{\Gamma}_1^T\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{P}_i^T\mathbf{\Gamma}_1^T\mathbf{\Gamma}_1\mathbf{P}_i)\mathbf{X}. \end{aligned} \tag{20}$$

In our example, this yields a highly structured RSM, yet does not reflect the task structure, *i.e.*, the hierarchical relationships between items (Figure 3C). Accordingly, we make

Definition 3.10. Neural representations whose RSM depends on the specific choice of input weights are *task-agnostic* representations.

In contrast, hidden-layer representations of MRNS

$$\mathbf{H} = \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+\mathbf{X} \tag{21}$$

and MWNS

$$\mathbf{H} = \mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T\mathbf{X} \tag{22}$$

are unique up to an orthogonal transformation $\mathbf{R}$, which includes rotations and reflections. Thus, in both cases, hidden-layer representations are not unique. In the semantic hierarchy task, this results in representations that appear arbitrary and unstructured (Figure 3D,E). However,

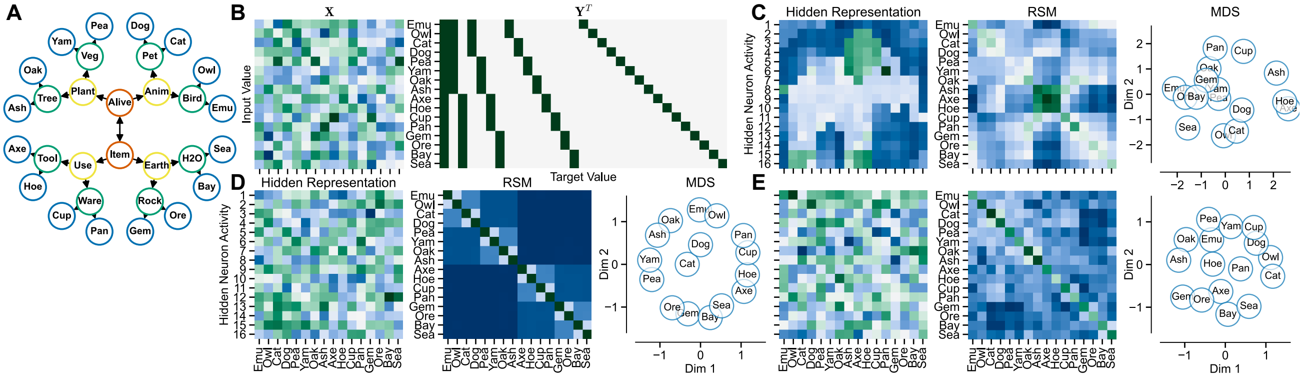

Figure 3: Hidden-layer representations. **(A)** Schematic of the semantic hierarchy task. **(B)** Inputs are encoded as random vectors (left) and corresponding target vectors encode for the position in the hierarchy (right). A one (zero) indicates that an item is (not) a child of a node. **(C)** Example hidden-layer representations (left), RSM (center) and corresponding 2D multidimensional scaling plot (right) for a GLS, **(D)** MRNS, and **(E)** MWNS.

Corollary 3.11. *The RSM of MRNS is unique and given by*

$$\mathrm{RSM} = \mathbf{O}\mathbf{N}\mathbf{O}^T. \tag{23}$$

Since $\mathbf{O}$ and $\mathbf{N}$ are fully determined by the training data, the RSM is invariant to the specific choice of input weights. Similarly,

Corollary 3.12. *The RSM of MWNS is unique and given by*

$$\mathrm{RSM} = \mathbf{X}^T\mathbf{V}\mathbf{S}\mathbf{V}^T\mathbf{X}. \tag{24}$$

Again, the RSM is invariant to the specific choice of input weights, as $\mathbf{V}$ and $\mathbf{S}$ are fully determined by the training data. Accordingly, we make

Definition 3.13. Neural representations whose RSM if fully determined by the training data are *task-specific* representations.

In the semantic hierarchy task, this yields an RSM that reflects the hierarchical structure, where representational similarity increases with proximity in the hierarchy, for MRNS; and an RSM that reflects a combination of input statistics and target hierarchy for MWNS (Figure 3D,E).

In summary, in two-layer linear networks, neural representations and the underlying function are dissociable: the same function can arise from different hidden-layer representations, and the same representations can support different functions. For GLS and LSS this flexibility supports almost arbitrary representations and task-agnostic RSMs. In contrast, MRNS and MWNS impose constraints which determine representations up to orthonormal transformations, and are guaranteed to have unique and task-specific RSMs.

4. Implications for neural data analysis

A fundamental challenge in understanding computations and learning in artificial and biological neural networks is linking changes in connectivity and representations to changes in function. However, if representation and function are dissociable, two key observations follow. First, changes in network function need not alter neural representations, as changes in one layer can be offset by compensatory changes in the next. Second, changes in network function can proceed without altering representations, as it can, in principle, occur entirely in downstream layers. We now examine the implications of these findings for representational comparisons, representational drift, and the synaptic stability-plasticity trade-off through a series of illustrative simulations. While exact outcomes depend on task specifics and hyperparameter choices, these are not our focus here; a systematic analytical and numerical exploration is left for future work.

4.1. Linear predictivity

A common method for comparing neural representations is to assess how well activation patterns from a source model or recording can predict those of a target via linear regression (*e.g.*, Yamins et al. 2014; Yamins and DiCarlo 2016). High linear predictivity is often interpreted as evidence that two systems process information similarly. However, since function and hidden-layer representations are dissociable, strong linear predictivity does not necessarily imply functional alignment. We illustrate this by comparing hidden-layer representations from independent random walks on the solution manifold of LSS, MWNS, and MRNS (Figure 4A). In each case, we compare either representations from two different network functions (across function) or from the same network function (within function). While R^2 scores are on average slightly higher for within-function comparisons, the main determinant of predictivity is whether the representations are task-agnostic or task-specific. Specifically, in across-function comparisons, R^2 scores are highest when the source representation comes from a LSS (which shares representational degrees of freedom with GLS), fol-

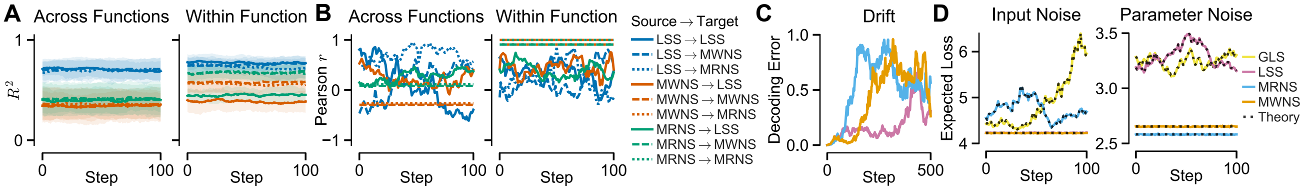

Figure 4: Implications for neural data analysis. All panels show results during random walks on the solution manifolds of LSS, MWNS, and MRNS. **(A)** Mean and standard deviation of R^2 scores for linear predictivity across $n = 10$ random walks. All source-target combinations are shown for across-function (left) and within-function (right) comparisons. **(B)** Example trajectories of RSA correlation scores, shown for across-function (left) and within-function (right) comparisons. **(C)** MSE of a linear decoder trained on the hidden-layer representation at the initial time step. **(D)** Mean and expected MSE under input noise (left) and parameter noise (right).

lowed by MRNS, and lowest for MWNS. Within-function comparisons yield high R^2 when the source is task-agnostic or when both source and target are of the same type; in contrast, predicting a task-agnostic representation from a task-specific one lead to the lowest R^2 scores. These patterns arise because task-agnostic solutions process both relevant and irrelevant input directions, producing higher-rank representations that cannot be linearly predicted from the lower-rank task-specific ones. These results indicate that linear predictivity is predominantly driven by solution type rather than functional alignment, and may yield misleading conclusions if underlying representational constraints are not explicitly taken into account.

4.2. Representational similarity analysis

Representational similarity analysis (RSA) compares neural activation patterns by evaluation the similarity of RSMs across conditions, stimuli, models, or participants (Kriegeskorte et al. 2008; Haxby et al. 2014). As with linear predictivity, our analytical results show that the interpretability of RSA in terms of functional alignment critically depends on the solution type of the comparanda. We illustrate this in Figure 4B using example trajectories from the previous section. Since task-agnostic solutions (*i.e.*, GLS and LSS) exhibit highly flexile RSMs, correlation coefficients r involving such representations fluctuate unpredictably throughout the random walk, both within and across functions. By contrast, comparing within and across task-specific solutions (*i.e.*, MWNS and MRNS) results in static and consistent r, as they exhibit unique RSMs. However, because MWNS and MRNS induce different unique RSMs, comparisons across types yield imperfect similarity even within the same function. In summary, RSA reliably reflects functional similarity only when representational constraints enforce unique RSMs, underscoring the importance of accounting for solution type in representational comparisons.

4.3. Drifting neural representations

Intuitively, one might expect that if a stimulus elicits stable perception and behaviour, the associated neural representations should likewise remain stable (Rule et al. 2019; Driscoll et al. 2022). However, this assumptions is challenged by converging evidence of representation drift across species, brain regions and modalities (*e.g.*, Ziv et al. 2013; Driscoll et al. 2017; Schoonover et al. 2021; Marks and Goard 2021; Deitch et al. 2021; Alisha et al. 2023). Our analysis shows that the existence of a solution manifold allows hidden-layer representations to vary without changing the implemented function. This dissociation implies that stable perception and behaviour do not require stable representations. Indeed, an optimal linear decoder trained on an initial representation rapidly degrades in performance during a random walk on the solution manifold (Figure 4C). Thus, representational drift need not signal functional change, but may instead reflect a reparameterisation within a functionally equivalent subspace.

4.4. Synaptic stability and plasticity

The so-called stability-plasticity dilemma posits that neural systems must remain plastic enough to acquire new knowledge while remaining stable enough to retrain previously learned information (Grossberg 1987; Abraham and Robins 2005). This view, grounded in single-neuron and synapse-level intuitions, has motivated continual learning algorithms that explicitly regulate synaptic changes to preserve past knowledge (*e.g.*, Kirkpatrick et al. 2017; Zenke et al. 2017; Aljundi et al. 2018). However, our analysis shows that this dilemma need not apply at the network level, because, independent of the solution type, many distinct configurations of synapses and representations implement the same function (see *e.g.*, Figure 1). This raises important methodological concerns: observing or inducing isolated synaptic or representational changes may not suffice to infer function, learning, or the underlying learning mechanisms in artificial or biological neural networks.

5. Advantages of task-specific representations

A natural question arises from the observation that function and representation are dissociable: why do biological and artificial systems often converge to non-arbitrary representations that obey the structure of the task? One possible explanation is that task-specific representations confer computational advantages, creating selective pressure in both biological and artificial systems. Consequently, such representations may emerge as preferred functional implementations, leading to representational alignment across systems.

5.1. Secondary error

One hypothesised advantage of task-specific representations is improved performance on secondary datasets, such as in- or out-of-distribution generalisation. To test this, we identify solutions on the primary-task solution manifold that minimise the error on an unseen secondary dataset $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{x}}_n, \tilde{\mathbf{y}_n})\}_{n=1}^{Q}$. In Appendix F we derive,

Theorem 5.1. *The secondary error is minimised by all solutions of the form*

$$\mathbf{\Omega}_2\mathbf{\Omega}_1 = \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \tilde{\mathbf{Z}}\mathbf{P}_u, \tag{25}$$

where

$$\tilde{\mathbf{Z}} = \left(\tilde{\mathbf{Y}} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{X}}\right)\left(\mathbf{P}_u\tilde{\mathbf{X}}\right)^{+} + \tilde{\mathbf{\Gamma}}\tilde{\mathbf{P}}_u, \tag{26}$$

with $\tilde{\mathbf{\Gamma}} \in \mathcal{R}^{N_o \times N_i}$ arbitrary and $\tilde{\mathbf{P}}_u = \mathbf{I} - \mathbf{P}_u\tilde{\mathbf{X}}(\mathbf{P}_u\tilde{\mathbf{X}})^{+}$.

Since the solution is a LSS with a perturbation in the unobserved null directions of the primary-task inputs, minimising secondary error permits task-agnostic solutions. Hence, observing $\mathbf{H}$ gives no information about secondary task performance and secondary error alone cannot explain the emergence of task-specific representations in two-layer linear networks.

5.2. Sensitivity to noise

Neural systems are subject to a multitude of internal and external sources of noise, which range from variability in incoming sensory signals to fluctuations in synaptic efficacy and spontaneous neural activity (Faisal et al. 2008). Therefore, solutions that exhibit robustness to such noise are advantageous, as they enable the neural circuitry to maintain reliable function (Johnston et al. 2020). The following theorems are derived in Appendix G.

Theorem 5.2. *The expected loss under additive, independent and identically distributed (i.i.d.), zero-centred input noise $\boldsymbol{\xi}_{\mathbf{x}_n}$ with variance σ_x^2 is*

$$\begin{aligned}&\left\langle \frac{1}{2P}\sum_{n=1}^{P} ||\mathbf{\Omega}_2\mathbf{\Omega}_1\left(\mathbf{x}_n + \boldsymbol{\xi}_{\mathbf{x}_n}\right) - \mathbf{y}_n||_2^2 \right\rangle \\ &= \frac{\sigma_{\mathbf{x}}^2}{2}||\mathbf{\Omega}_2\mathbf{\Omega}_1||_F^2 + c,\end{aligned} \tag{27}$$

where c is a noise-independent constant that only depends on the training data.

Corollary 5.3. *Any LSS or MRNS minimises the expected loss under input noise (see Figure 4D).*

Thus, while robustness to input noise selects for solutions with minimal functional norm, LSS employ task-agnostic representations, so robustness alone does not ensure representational alignment.

Theorem 5.4. *The expected loss under additive, i.i.d., zero-centred noise Ξ_1 and Ξ_2 in the parameters, with variances $\sigma_1^2 \propto 1/||\mathbf{X}||_F^2$ and $\sigma_2^2 \propto 1/N_o$ is*

$$\begin{aligned}&\left\langle \frac{1}{2P}\sum_{n=1}^{P} ||\left(\mathbf{\Omega}_2 + \mathbf{\Xi}_2\right)\left(\mathbf{\Omega}_1 + \mathbf{\Xi}_1\right)\mathbf{x}_n - \mathbf{y}_n||_2^2 \right\rangle \\ &= \frac{1}{2P}\Big(||\mathbf{\Omega}_1\mathbf{X}||_F^2 + ||\mathbf{\Omega}_2||_F^2 + c\Big).\end{aligned} \tag{28}$$

where c is again a noise-independent constant.

Corollary 5.5. *Any MRNS minimises the expected loss under parameter noise (see Figure 4D).*

We have scaled noise variances to simplify the analytical expression; without this scaling, the results hold up to multiplicative constants. Robustness to parameter noise thus selects for solutions with task-specific representations, ensuring representational alignment.

Theorem 5.6. *Under the assumption that the input data is whitened, i.e., $\mathbf{\Sigma}_{xx} = \mathbf{I}$, the expected loss under additive, i.i.d., zero-centred parameter noise Ξ_1 and Ξ_2 with variance $\sigma_1^2 \propto 1/N_i$ and $\sigma_2^2 \propto 1/N_o$ and input noise $\boldsymbol{\xi}_{\mathbf{x}_n}$ with variance σ_x^2 is*

$$\begin{aligned}&\left\langle \frac{1}{2P}\sum_{n=1}^{P} ||\left(\mathbf{\Omega}_2 + \mathbf{\Xi}_2\right)\left(\mathbf{\Omega}_1 + \mathbf{\Xi}_1\right)\left(\mathbf{x}_n + \boldsymbol{\xi}_{\mathbf{x}_n}\right) - \mathbf{y}_n||_2^2 \right\rangle \\ &= \frac{\sigma_{\mathbf{x}}^2}{2}\left(||\mathbf{\Omega}_2\mathbf{\Omega}_1||_F^2 + ||\mathbf{\Omega}_2||_F^2 + ||\mathbf{\Omega}_1||_F^2\right) \\ &\qquad + \frac{1}{2P}\left(||\mathbf{\Omega}_2||_F^2 + ||\mathbf{\Omega}_1\mathbf{X}||_F^2\right) + c,\end{aligned} \tag{29}$$

with noise-independent constant c.

Corollary 5.7. *Under the stated assumptions and constraints, any MRNS or MWNS minimises the expected loss under input and parameter noise.*

We note, that if the input data is not whitened, interaction terms render the optimal subspace depends on input statistics, complicating its explicit analytical characterisation.

In summary, neither minimising secondary error nor input noise sensitivity promotes task-specific representations. In contrast, robustness to parameter noise selectively favours solutions with task-specific structure, thereby supporting representational alignment. This suggests that shared representations across biological and artificial systems may arise from implicit or explicit optimisation for parameter

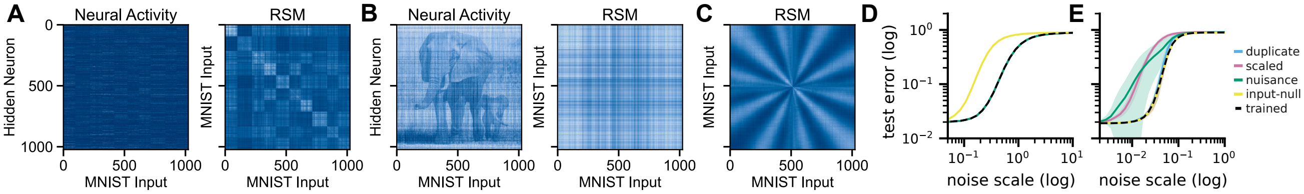

Figure 5: Function and representation are dissociable in non-linear networks. **(A)** Hidden-layer activations for 1024 MNIST inputs, grouped by class (left) and the corresponding task-specific RSM (right) after training a ReLU network from small initial weights. **(B)** Same as **(A)** , but for a network reparameterised via augmented Lagrangian optimisation to reshape hidden-layer representation while preserving training set classifications. **(C)** RSM of the network from **(A)** after reparameterisation using exact invariant transformations (Section 6.1). **(D)** Test error under input noise for all exact invariant transformations. Networks with `input-null` expansion are sensitive. **(E)** As in **(D)** but for parameter noise; networks with `scaled`, `nuisance`, and `duplicate` expansions are sensitive.

robustness, such as regularisation strategies that favour low-norm solutions. However, this reflects an inductive bias rather than a general principle and without explicit justification, functional and representational alignment cannot be assumed to coincide.

6. Nonlinear networks

The results presented thus far apply to two-layer linear networks. We now extend our study to emphnonlinear networks (networks with nonlinear activation functions), and show that they exhibit analogous degrees of freedom in representation and associated computational trade-offs as their linear counterparts. Here, we face a challenge: A full characterisation of the solution manifold of general nonlinear network remains analytically intractable, even for two-layer networks (Misiakiewicz and Montanari 2024). However, substantial progress has been made in deriving function-preserving transformations that allow reparametrisation of nonlinear networks while leaving their input-output map invariant (Simsek et al. 2021; Martinelli et al. 2023). Here, we exploit these function-preserving transformations to construct functionally equivalent reparametrisations of nonlinear networks, which allows us to probe computational differences between networks with minimal and expanded representations, in analogy to the task-specific and task-agnostic representations of Section 3.

6.1. Functional invariances in deep ReLU networks

Feedforward networks with any activation function are output-invariant to permuting neurons within a layer, as permuting the rows of one weight matrix and the corresponding columns of the next leaves the network function unchanged (*permutation invariance*, Sussmann 1992). Feedforward networks with rectified linear unit (ReLU) activation are further invariant to rescaling a neuron's incoming weights by a factor $\alpha > 0$ and its outgoing weights by $1/\alpha$, due to the non-negative homogeneity of ReLU (*scale invariance*, Neyshabur et al. 2015a). Simsek et al. (2021) and Martinelli et al. (2023) identify additional invariances that fully characterise the manifold of global minima in teacher-student settings, where one network is trained to replicate the function of another. Although these invariances may not capture all functionally equivalent parametrisations outside of the teacher-student setting, they provide a means to construct network reparametrisations that exactly preserve network function. We realize two of their invariances by inserting hidden-layer neurons with arbitrary incoming weights and zero outgoing weights (*nuisance-neuron invariance*) and by duplicating hidden-layer neurons while halving the outgoing weights of both the original and the duplicated neuron (*duplication invariance*). Lastly, one can add any perturbation to the input-layer weights that lies in the unobserved nullspace of the input data while preserving the network's function (*input-nullspace invariance*).

6.2. Manipulating representations of ReLU networks

We train a two-layer ReLU network with 1024 hidden neurons on the MNIST dataset (LeCun et al. 1998) from small norm random weights, resulting in task-specific representations (Figure 5A, "rich" learning from Jacot et al. 2018; Chizat et al. 2019; Woodworth et al. 2020). Starting from this trained model, we use augmented Lagrangian optimisation to modify the hidden-layer activations of 1024 training inputs such that their representations collectively resemble an image of two elephants, while enforcing that the network's predicted class labels remain unchanged across the entire training set (Figure 5B). This transformation illustrates the extensive representational freedom in ReLU networks. Although class predictions remain fixed in this case, the underlying network function and its decision boundaries change, as classification ignores relative differences. However, even when limited to exact invariances, one can induce nearly arbitrary RSMs (Figure 5C).

6.3. Computational advantages in ReLU networks

To complement the analytical results in Section 5.2 on the robustness of task-specific representations in linear networks, we empirically evaluate secondary (test) error and robustness to input and parameter noise across different nonlinear solution types. We train two-layer networks with 1024 hidden dimensions and ReLU activation on the training set of the MNIST digit classification task (LeCun et al. 2010) from 8 random initialisations. These models trained from small initial weights serve our *task-trained* (minimal) solutions. We next apply four function-preserving transforms defined by the four invariances of Section 6.1 to these task-trained (minimal) networks to construct expanded (non-minimal) parametrisations that exactly preserve the network function. These minimal and non-minimal parametrisations serve as our solution types in the nonlinear setting.

In Figure 5D, we observe the effect of **input noise** on the task-trained (minimal) and expanded (non-minimal) networks. In accordance with the linear result, only transformations in the unobserved input space have a deleterious effect (`input-null`). In Figure 5E, we observe the effect of **parameter noise** on the initial and transformed networks. Non-minimal models (`scaled`, `nuisance`, `duplicate`) degrade in test error more quickly than minimal ones, similarly to what is derived in Section 5.2, though duplicated expansions (`duplicate`) are more robust due to noise averaging. In contrast, input-nullspace perturbations (`input-null`) have no effect because transformations are in unobserved input directions, and input noise is absent. Lastly, at near-zero noise levels, we observe that no manipulations inflate the secondary (test) error, consistent with the result of Section 5.1 that generalisation performance does not constrain network representations to be minimal.

7. Related work

The solution manifold of artificial networks. The solution manifold of two-layer neural networks was first described by Baldi and Hornik (1989). Subsequent work showed under some assumptions that all minima in deep linear networks are global and equivalent to those in linear regression (Laurent and Brecht 2018). The dissociation between general linear solutions and minimum-norm solutions has been previously studied under a set of strong assumptions (Saxe et al. 2014). Sensitivity to noise for task-specific and task-agnostic rich and lazy solutions has been previously studied numerically in nonlinear neural networks (Flesch et al. 2022) and generalisation and transfer performance of deep linear networks have been previously investigated (Lampinen and Ganguli 2019; Advani et al. 2020; Tahir et al. 2024; Ingrosso et al. 2024) using a teacher-student paradigm (Gardner and Derrida 1989; Riegler and Biehl 1995; Saad and Solla 1995). The relation between representational drift and drift on the solution manifold that results from stochasticity during gradient descent (Chaudhari and Soatto 2018) has been studied on a subpart of the solution manifold in linear networks (Pashakhanloo and Koulakov 2023) and in nonlinear networks, again, relying on the teacher-student setting (Avidan et al. 2023; Li et al. 2024).

Comparing the solutions of artificial and biological networks. Neuroscientists have identified parallels between artificial and biological neural computation (Richards et al. 2019; Saxe et al. 2021; Doerig et al. 2023) from hierarchical feature extraction in visual processing (DiCarlo et al. 2012; Eickenberg et al. 2017; Lindsay 2021) to analogous population dynamics during decision-making tasks (Mante et al. 2013; Chaisangmongkon et al. 2017). The field has developed various methods to quantify this shared representational structure, including representational similarity analysis (Kriegeskorte et al. 2008), linear predictivity (*e.g.*, Yamins et al. 2014; Yamins and DiCarlo 2016), and metric-based methods (Williams et al. 2021); see Klabunde et al. (2023) for an overview of methods. However, recent work has identified significant methodological challenges in comparing representations between artificial neural networks, including confounding effects from stimulus correlations (Cai et al. 2019; Hermann and Lampinen 2020; Dujmović et al. 2023), metric-dependent results (Ding et al. 2021; Soni et al. 2024), and difficulties in matching representations even between identical networks trained with different random initialisations (Han et al. 2023); these negative results suggest further problems when comparing artificial and biological neural networks, where little is known in advance of the computation the biological networks performs. These challenges may apply to existing neural predictivity benchmarks such as Brain-score (Schrimpf et al. 2020) and the Natural Scenes Dataset (NSD; Allen et al. 2022).

8. Discussion

In this work, we give a complete analytical characterisation of the global minima manifold for deep linear networks, and demonstrate that different subregions of this manifold afford different interpretability and computational properties due to their representational structure. We conclude that the use of deep, overparametrised networks poses fundamental challenges for representational analysis, interpretation, and comparison, as the impact of variability in the parametrisation of functionally equivalent representations on these use cases is significant.

Our analysis does not assume a specific learning algorithm and ignores the question of how a specific solution could be attained in practice. However, the computational advantages of task-specific representations detailed in Section 5 are compatible with the view that gradient descent, in particular in overparametrised models, is subject to *implicit regularisation* that prefers solutions with certain optimality properties for both linear and nonlinear networks (Zhang et al. 2017; Yun et al. 2021; Vardi and Shamir 2021; Neyshabur 2017; Neyshabur et al. 2015b; Du et al. 2019; Arora et al. 2018; Chizat and Bach 2020).

Acknowledgements

L.B. was supported by the Woodward Scholarship awarded by Wadham College, Oxford and the Medical Research Council [MR/N013468/1]. E.G. and A.S. were supported by the Gatsby Charitable Foundation (GAT3850). Further, A.S. was supported by a Sir Henry Dale Fellowship from the Wellcome Trust and Royal Society (216386/Z/19/Z) and the Sainsbury Wellcome Centre Core Grant (219627/Z/19/Z).

Impact statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Abraham, Wickliffe C and Anthony Robins (2005). “Memory retention–the synaptic stability versus plasticity dilemma”. In: *Trends in neurosciences* 28.2, pp. 73–78

Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky (2020). “High-dimensional dynamics of generalization error in neural networks”. In: *Neural Networks* 132, pp. 428–446

Albantakis, Larissa et al. (2024). “The brain’s best kept secret is its degenerate structure”. In: *Journal of Neuroscience* 44.40

Albertini, Francesca and Eduardo D. Sontag (1993). “For neural networks, function determines form”. In: *Neural Networks* 6.7, pp. 975–990

Alisha, Ahmed, Voelcker Bettina, and Peron Simon (2023). “Representational drift in barrel cortex is receptive field dependent”. In: *bioRxiv*

Aljundi, Rahaf et al. (2018). “Memory aware synapses: Learning what (not) to forget”. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154

Allen, Emily J. et al. (2022). “A massive 7T fMRI dataset to bridge cognitive neuroscience and artificial intelligence”. In: *Nature Neuroscience* 25.1, pp. 116–126

Arora, Sanjeev, Nadav Cohen, Noah Golowich, and Wei Hu (2019). “A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks”. In: *7th International Conference on Learning Representations*. OpenReview.net

Arora, Sanjeev, Nadav Cohen, and Elad Hazan (2018). “On the optimization of deep networks: Implicit acceleration by overparameterization”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Pmlr, pp. 244–253

Arous, Gérard Ben, Reza Gheissari, and Aukosh Jagannath (2022). “High-dimensional limit theorems for SGD: Effective dynamics and critical scaling”. In: *Advances in Neural Information Processing Systems 35*. Ed. by Sanmi Koyejo et al.

Avidan, Yehonatan, Qianyi Li, and Haim Sompolinsky (2023). *Connecting NTK and NNGP: A Unified Theoretical Framework for Wide Neural Network Learning Dynamics*.

Baldi, Pierre and Kurt Hornik (Jan. 1, 1989). “Neural Networks and Principal Component Analysis: Learning from Examples without Local Minima”. In: *Neural Networks* 2.1, pp. 53–58

Bona-Pellissier, Joachim, François Bachoc, and François Malgouyres (2021). *Parameter identifiability of a deep feedforward ReLU neural network*

Braun, Lukas, Clémentine Dominé, James Fitzgerald, and Andrew Saxe (2022). “Exact learning dynamics of deep linear networks with prior knowledge”. In: *Advances in Neural Information Processing Systems 35*. Ed. by Sanmi Koyejo et al.

Cai, Ming Bo, Nicolas W. Schuck, Jonathan W. Pillow, and Yael Niv (May 2019). “Representational Structure or Task Structure? Bias in Neural Representational Similarity Analysis and a Bayesian Method for Reducing Bias”. In: *PLOS Computational Biology* 15.5, pp. 1–30

Chaisangmongkon, Warasinee, Sruthi K. Swaminathan, David J. Freedman, and Xiao-Jing Wang (Mar. 22, 2017). “Computing by Robust Transience: How the Fronto-Parietal Network Performs Sequential, Category-Based Decisions”. In: *Neuron* 93.6, 1504–1517.e4

Chaudhari, Pratik and Stefano Soatto (2018). “Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks”. In: *Proceedings of the 6th International Conference on Learning Representations*. OpenReview.net

Chizat, Lénaïc and Francis R. Bach (2020). “Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss”. In: *Conference on Learning Theory*. Ed. by Jacob D. Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. Pmlr, pp. 1305–1338

Chizat, Lénaïc, Edouard Oyallon, and Francis R. Bach (2019). “On Lazy Training in Differentiable Programming”. In: *Advances in Neural Information Processing Systems*. Ed. by Hanna M. Wallach et al., pp. 2933–2943

Deitch, Daniel, Alon Rubin, and Yaniv Ziv (2021). “Representational drift in the mouse visual cortex”. In: *Current biology* 31.19, pp. 4327–4339

DiCarlo, James J., Davide Zoccolan, and Nicole C. Rust (Feb. 9, 2012). “How Does the Brain Solve Visual Object Recognition?” In: *Neuron* 73.3, pp. 415–434

Ding, Frances, Jean-Stanislas Denain, and Jacob Steinhardt (2021). “Grounding Representation Similarity Through Statistical Testing”. In: *Advances in Neural Information Processing Systems 34*. Ed. by Marc’Aurelio Ranzato et al., pp. 1556–1568

Doerig, Adrien et al. (2023). “The neuroconnectionist research programme”. In: *Nature Reviews Neuroscience* 24.7 (7), pp. 431–450

Dominé, Clémentine CJ et al. (2024). “From lazy to rich: Exact learning dynamics in deep linear networks”. In: *arXiv preprint arXiv:2409.14623*

Driscoll, Laura N, Lea Duncker, and Christopher D Harvey (2022). “Representational drift: Emerging theories for continual learning and experimental future directions”. In: *Current Opinion in Neurobiology* 76, p. 102609

Driscoll, Laura N et al. (2017). “Dynamic reorganization of neuronal activity patterns in parietal cortex”. In: *Cell* 170.5, pp. 986–999

Du, Simon S., Xiyu Zhai, Barnabás Póczos, and Aarti Singh (2019). “Gradient Descent Provably Optimizes Over-parameterized Neural Networks”. In: *7th International Conference on Learning Representations*. OpenReview.net

Dujmović, Marin, Jeffrey S. Bowers, Federico Adolfi, and Gaurav Malhotra (2023). *Obstacles to inferring mechanistic similarity using representational similarity analysis*

Eickenberg, Michael, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion (May 15, 2017). “Seeing It All: Convolutional Network Layers Map the Function of the Human Visual System”. In: *NeuroImage* 152, pp. 184–194

Entezari, Rahim, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur (2022). “The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks”. In: *The 10th International Conference on Learning Representations*. OpenReview.net

Faisal, A Aldo, Luc PJ Selen, and Daniel M Wolpert (2008). “Noise in the nervous system”. In: *Nature reviews neuroscience* 9.4, pp. 292–303

Flesch, Timo et al. (Apr. 6, 2022). “Orthogonal Representations for Robust Context-Dependent Task Performance in Brains and Neural Networks”. In: *Neuron* 110.7 (7), 1258–1270.e11

Gardner, E. and B. Derrida (1989). “Three unfinished works on the optimal storage capacity of networks”. In: *Journal of Physics A: Mathematical and General* 22.12, p. 1983

Ghosh, Shubhangi et al. (2022). *On Pitfalls of Identifiability in Unsupervised Learning. A Note on: ”Desiderata for Representation Learning: A Causal Perspective”.*

Godfrey, Charles, Davis Brown, Tegan Emerson, and Henry Kvinge (2022). “On the symmetries of deep learning models and their internal representations”. In: *Advances in Neural Information Processing Systems* 35, pp. 11893–11905

Goldman, Mark S, Jorge Golowasch, Eve Marder, and LF Abbott (2001). “Global structure, robustness, and modulation of neuronal models”. In: *Journal of Neuroscience* 21.14, pp. 5229–5238

Greville, Thomas Nall Eden (1966). “Note on the generalized inverse of a matrix product”. In: *Siam Review* 8.4, pp. 518–521

Grossberg, Stephen (1987). “Competitive learning: From interactive activation to adaptive resonance”. In: *Cognitive science* 11.1, pp. 23–63

Han, Yena, Tomaso A. Poggio, and Brian Cheung (2023). “System identification of neural systems: If we got it right, would we know?” In: *International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. Pmlr, pp. 12430–12444

Haxby, James V, Andrew C Connolly, and J Swaroop Guntupalli (2014). “Decoding neural representational spaces using multivariate pattern analysis”. In: *Annual review of neuroscience* 37.1, pp. 435–456

Hermann, Katherine L. and Andrew K. Lampinen (2020). “What shapes feature representations? Exploring datasets, architectures, and training”. In: *Advances in Neural Information Processing Systems 33*. Ed. by Hugo Larochelle et al.

Ingrosso, Alessandro, Rosalba Pacelli, Pietro Rotondo, and Federica Gerace (2024). *Statistical Mechanics of Transfer Learning in Fully-Connected Networks in the Proportional Limit.*

Jacot, Arthur, Franck Gabriel, and Clément Hongler (2018). “Neural tangent kernel: Convergence and generalization in neural networks”. In: *Advances in neural information processing systems* 31

Johnston, W Jeffrey, Stephanie E Palmer, and David J Freedman (2020). “Nonlinear mixed selectivity supports reliable neural computation”. In: *PLoS computational biology* 16.2, e1007544

Kirkpatrick, James et al. (2017). “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526

Klabunde, Max, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich (2023). *Similarity of neural network models: A survey of functional and representational measures*

Kori, Avinash et al. (2024). *Identifiable object-centric representation learning via probabilistic slot attention.*

Kriegeskorte, Nikolaus, Marieke Mur, and Peter A Bandettini (2008). “Representational similarity analysis - connecting the branches of systems neuroscience”. In: *Frontiers in systems neuroscience* 2, p. 249

Lampinen, Andrew K. and Surya Ganguli (2019). “An analytic theory of generalization dynamics and transfer learning in deep linear networks”. In: *7th International Conference on Learning Representations*. OpenReview.net

Laurent, Thomas and James von Brecht (2018). “Deep linear networks with arbitrary loss: All local minima are global”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Pmlr, pp. 2908–2913

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324

LeCun, Yann, Corinna Cortes, and CJ Burges (2010). “MNIST handwritten digit database”. In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2

Li, Qianyi, Ben Sorscher, and Haim Sompolinsky (2024). “Representations and generalization in artificial and brain neural networks”. In: *Proceedings of the National Academy of Sciences* 121.27, e2311805121

Lindsay, Grace W. (Sept. 1, 2021). “Convolutional Neural Networks as a Model of the Visual System: Past, Present, and Future”. In: *Journal of Cognitive Neuroscience* 33.10, pp. 2017–2031

Mante, Valerio, David Sussillo, Krishna V. Shenoy, and William T. Newsome (Nov. 2013). “Context-Dependent Computation by Recurrent Dynamics in Prefrontal Cortex”. In: *Nature* 503.7474 (7474), pp. 78–84

Marconato, Emanuele, Sébastien Lachapelle, Sebastian Weichwald, and Luigi Gresele (2024). *All or None: Identifiable Linear Properties of next-Token Predictors in Language Modeling*.

Marder, Eve and Jean-Marc Goaillard (2006). “Variability, compensation and homeostasis in neuron and network function”. In: *Nature Reviews Neuroscience* 7.7, pp. 563–574

Marks, Tyler D and Michael J Goard (2021). “Stimulus-dependent representational drift in primary visual cortex”. In: *Nature communications* 12.1, p. 5169

Martinelli, Flavio, Berfin Şimşek, Johanni Brea, and Wulfram Gerstner (2023). *Expand-and-Cluster: Exact Parameter Recovery of Neural Networks*.

Misiakiewicz, Theodor and Andrea Montanari (Oct. 2024). “Six Lectures on Linearized Neural Networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2024.10, p. 104006

Neyshabur, Behnam (2017). “Implicit Regularization in Deep Learning”

Neyshabur, Behnam, Ruslan Salakhutdinov, and Nathan Srebro (2015a). “Path-SGD: Path-Normalized Optimization in Deep Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by Corinna Cortes et al., pp. 2422–2430

Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro (Apr. 16, 2015b). “In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning”. In: *ICLR 2015 Workshop Track*. International Conference on Learning Representations

Pashakhanloo, Farhad and Alexei Koulakov (2023). “Stochastic gradient descent-induced drift of representation in a two-layer neural network”. In: *International Conference on Machine Learning*. Ed. by Andreas Krause et al. Vol. 202. Proceedings of Machine Learning Research. Pmlr, pp. 27401–27419

Prinz, Astrid A, Dirk Bucher, and Eve Marder (2004). “Similar network activity from disparate circuit parameters”. In: *Nature neuroscience* 7.12, pp. 1345–1352

Refinetti, Maria, Sebastian Goldt, Florent Krzakala, and Lenka Zdeborová (2021). “Classifying high-dimensional Gaussian mixtures: Where kernel methods fail and neural networks succeed”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. Pmlr, pp. 8936–8947

Richards, Blake A. et al. (2019). “A deep learning framework for neuroscience”. In: *Nature Neuroscience* 22.11 (11), pp. 1761–1770

Riegler, P. and M. Biehl (Oct. 1995). “On-Line Backpropagation in Two-Layered Neural Networks”. In: *Journal of Physics A: Mathematical and General* 28.20 (20), p. L507

Roeder, Geoffrey, Luke Metz, and Durk Kingma (2021). “On Linear Identifiability of Learned Representations”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. Pmlr, pp. 9030–9039

Rule, Michael E, Timothy O’Leary, and Christopher D Harvey (2019). “Causes and consequences of representational drift”. In: *Current opinion in neurobiology* 58, pp. 141–147

Rumelhart, David E, James L McClelland, and the PDP Research Group (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. Cambridge, MA: MIT Press

Saad, David and Sara A. Solla (Oct. 1, 1995). “On-Line Learning in Soft Committee Machines”. In: *Physical Review E* 52.4 (4), pp. 4225–4243

Saxe, Andrew, Stephanie Nelli, and Christopher Summerfield (Jan. 2021). “If deep learning is the answer, what is the question?” In: *Nature Reviews Neuroscience* 22.1 (1), pp. 55–67

Saxe, Andrew M, James L McClelland, and Surya Ganguli (2019). “A mathematical theory of semantic development in deep neural networks”. In: *Proceedings of the National Academy of Sciences* 116.23, pp. 11537–11546

Saxe, Andrew M., James L. McClelland, and Surya Ganguli (2014). "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *2nd International Conference on Learning Representations*. Ed. by Yoshua Bengio and Yann LeCun

Schoonover, Carl E, Sarah N Ohashi, Richard Axel, and Andrew JP Fink (2021). "Representational drift in primary olfactory cortex". In: *Nature* 594.7864, pp. 541–546

Schrimpf, Martin et al. (2020). *Brain-score: Which artificial neural network for object recognition is most brain-like?*

Simsek, Berfin et al. (2021). "Geometry of the Loss Landscape in Overparameterized Neural Networks: Symmetries and Invariances". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. Pmlr, pp. 9722–9732

Soni, Ansh, Sudhanshu Srivastava, Konrad Kording, and Meenakshi Khosla (2024). *Conclusions about neural network to brain alignment are profoundly impacted by the similarity measure*.

Sussmann, Héctor J. (July 1, 1992). "Uniqueness of the Weights for Minimal Feedforward Nets with a given Input-Output Map". In: *Neural Networks* 5.4, pp. 589–593

Tahir, Javan, Surya Ganguli, and Grant M. Rotskoff (2024). *Features Are Fate: A Theory of Transfer Learning in High-Dimensional Regression*.

Vardi, Gal and Ohad Shamir (2021). "Implicit Regularization in ReLU Networks with the Square Loss". In: *Proceedings of 34th Conference on Learning Theory*. Ed. by Mikhail Belkin and Samory Kpotufe. Vol. 134. Proceedings of Machine Learning Research. Pmlr, pp. 4224–4258

Vlačić, Verner and Helmut Bölcskei (2022). "Neural network identifiability for a family of sigmoidal nonlinearities". In: *Constructive Approximation* 55.1, pp. 173–224

Wang, Yixin and Michael I. Jordan (2021). *Desiderata for Representation Learning: A Causal Perspective*.

Williams, Alex H., Erin Kunz, Simon Kornblith, and Scott W. Linderman (2021). "Generalized Shape Metrics on Neural Representations". In: *Advances in Neural Information Processing Systems 34*. Ed. by Marc'Aurelio Ranzato et al., pp. 4738–4750

Woodworth, Blake E. et al. (2020). "Kernel and Rich Regimes in Overparametrized Models". In: *Conference on Learning Theory*. Ed. by Jacob D. Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. Pmlr, pp. 3635–3673

Yamins, Daniel L K and James J DiCarlo (Mar. 2016). "Using Goal-Driven Deep Learning Models to Understand Sensory Cortex". In: *Nature Neuroscience* 19.3 (3), pp. 356–365

Yamins, Daniel L. K. et al. (2014). "Performance-Optimized Hierarchical Models Predict Neural Responses in Higher Visual Cortex". In: *Proceedings of the National Academy of Sciences* 111.23 (23), pp. 8619–8624

Yun, Chulhee, Shankar Krishnan, and Hossein Mobahi (2021). "A unifying view on implicit bias in training linear neural networks". In: *9th International Conference on Learning Representations*. OpenReview.net

Zenke, Friedemann, Ben Poole, and Surya Ganguli (2017). "Continual learning through synaptic intelligence". In: *International conference on machine learning*. Pmlr, pp. 3987–3995

Zhang, Chiyuan et al. (2017). "Understanding deep learning requires rethinking generalization". In: *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net

Zhang, Chiyuan et al. (2021). "Understanding deep learning (still) requires rethinking generalization". In: 64.3, pp. 107–115. ISSN: 0001-0782

Ziv, Yaniv et al. (2013). "Long-term dynamics of CA1 hippocampal place codes". In: *Nature neuroscience* 16.3, pp. 264–266

A. Simulation details

A.1. Random walk

Given a two-layer linear network with weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, we implemented random walks on the GLS as follows. First, we randomly initialised weight matrices using a random normal initialisation with zero mean and standard deviation $1/\sqrt{N_i}$ and $1/\sqrt{N_h}$ respectively. Then, for each step, we first diffused both weight matrices by

$$\mathbf{W}(t+1) = (1-\alpha)\mathbf{W}(t) + \alpha\xi \tag{30}$$

with

$$\xi \sim \mathcal{N}(\mu = 0, \sigma^2 = 25), \text{ and } \alpha = 0.00625 \tag{31}$$

and then performed gradient descent on both $\mathbf{W}_1$ and $\mathbf{W}_2$ with learning rate $\eta = 0.25$ to return back to the solution manifold. Similarly, for the random walk on the LSS, we sampled initial $\mathbf{W}_1$ and $\mathbf{W}_2$ randomly, diffused them, performed gradient descent on both $\mathbf{W}_1$ and $\mathbf{W}_2$, and subsequently enforced the two rank constraints to return to the solution manifold. The random walk on the MRNS was implemented by first computing

$$\text{cSVD}(\mathbf{Y}\mathbf{X}^T\mathbf{X}^{+T}) = \mathbf{M}\mathbf{N}\mathbf{O}^T, \tag{32}$$

sampling a random

$$\mathbf{\Gamma} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1/N_i), \quad \text{and} \quad \tilde{\mathbf{R}} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1). \tag{33}$$

Then, in every step, we diffused $\mathbf{\Gamma}$ and $\mathbf{R}$ according to Equation (30), and then calculated

$$\mathbf{R} = \mathbf{R}_1\mathbf{R}_2^T \quad \text{with} \quad \text{cSVD}(\tilde{\mathbf{R}}) = \mathbf{R}_1\mathbf{D}\mathbf{R}_2^T. \tag{34}$$

We then set

$$\mathbf{W}_1 \rightarrow \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+ + \mathbf{\Gamma} \quad \text{and} \quad \mathbf{W}_2 \rightarrow \mathbf{M}\sqrt{\mathbf{N}}\mathbf{R}^T. \tag{35}$$

Similarly, for MWNS we first computed the cSVD of the LSS

$$\text{cSVD}\left(\mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+\right) = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{36}$$

and sampled a random matrix $\tilde{\mathbf{R}} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1)$. Then, for each step, we defused $\tilde{\mathbf{R}}$ according to Equation (30) and then calculated $\mathbf{R}$ as for MWNS. Network weights were then set to

$$\mathbf{W}_1 \rightarrow \mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T \tag{37}$$

and

$$\mathbf{W}_2 \rightarrow \mathbf{U}\sqrt{\mathbf{S}}\mathbf{R}^T. \tag{38}$$

A.2. Tasks

Random regression task Random regression tasks are made up of a randomly sampled input

$$\mathbf{X} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1/N_i) \in \mathcal{R}^{N_i \times P} \tag{39}$$

and target matrix

$$\mathbf{Y} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1/N_o) \in \mathcal{R}^{N_o \times P}. \tag{40}$$

Semantic hierarchy Each of the $P = 16$ items in the semantic hierarchy task was encoded as a random normal distributed vector

$$\mathbf{X} \sim \mathcal{N}(\mu = 0, \sigma^2 = 1/N_i) \in \mathcal{R}^{N_i \times P}. \tag{41}$$

Corresponding target vectors were then generated according to the position of the item in the hierarchy. If an item is a child of a particular node it is encoded as a 1 and 0 otherwise. For example, a pea is alive, not an object, not an animal, a plant, not handy, not earth, not a bird, not furry, but a veg, not a tree ... resulting in the target vector [1, 0, 0, 1, 0, 0, 0, 0, 1, 0 ...].

A.3. Figure 1

Random walk on MRNS manifold of random regression task with $N_i = 4$, $N_h = 7$, $N_o = 3$, and $P = 3$ for 500 steps with diffusion parameter $\alpha = 0.005$.

A.4. Figure 3

The neural network has $N_i = 16$, $N_h = 16$, $N_o = 31$. The point on the LSS solution manifold which depicts an elephant $bfE \in \mathcal{R}^{N_h \times P}$, was found by first initialising the weight matrices as

$$\mathbf{W}_1 = \mathbf{E}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + 5 \times 10^{-3}\mathbf{I})^+ \quad \text{and} \quad \mathbf{W}_2 = \mathbf{Y}\mathbf{X}^T(\mathbf{W}_1\mathbf{X}(\mathbf{W}_1\mathbf{X})^T + 1 \times 10^{-2}\mathbf{I})^+, \tag{42}$$

and subsequently applying gradient descent on both weight matrices to find a point on the solution manifold. Randomly sampled points on the solution manifold of MRNS and MWNS were generated according to the definitions of $\mathbf{\Omega}_1$ in Theorem 3.5, and Theorem 3.7.

A.5. Figure 4

Random walk on solution manifold for GLS, LSS, MRNS, and MWNS on a random regression task with $N_i = 6$, $N_h = 16$, $N_o = 2$, and $P = 4$ for 100 steps.

(A) We fit a linear model to predict the hidden-layer activations of one model by the hidden-layer activations of another for each time step of the random walk, using ridge regression and subsequently calculated the R^2 score of that fit.

(B) We performed representational similarity analysis on pairwise random walk trajectories of (B) by calculating the euclidean distance matrix of their respective hidden-layer representations

$$\mathrm{RSM}_{ij} = ||\mathbf{h}_i - \mathbf{h}_j||_2 \tag{43}$$

and subsequently compared their off-diagonal values using Pearson correlation coefficients.

(C) To analyse drift, we fit a linear decoder on the hidden-layer representation of a model at the beginning of the random walk

$$\tilde{\mathbf{W}} = \mathbf{\Sigma}_{yx}\mathbf{W}_1^T(\mathbf{W}_1\mathbf{X}\mathbf{X}^T\mathbf{W}_1^T)^+ \tag{44}$$

and subsequently calculated the mean squared error of that classifier, given the hidden-layer representation at each time step.

(D) We numerically calculated the average loss across $n = 500000$ randomly sampled input noise vectors with $\sigma_{\mathbf{x}}^2 = 1$ at each time step. Accordingly, we numerically calculated the average loss across $n = 500000$ randomly sampled parameter noise matrices with $\sigma_1^2 = 1/||\mathbf{X}||_F^2$ and $\sigma_2^2 = 1/N_o$.

A.6. Simulating nonlinear networks

As a precursor to all numerical experiments with nonlinear networks in Section 6, we train feed-forward neural networks on a non-synthetic dataset to produce a *task-specific parametrisation*. All neural networks in this section are trained with back-propagation and (mini)-batch gradient descent from small initial weights, a regime that induces task-specific feature learning in nonlinear networks similarly to linear networks Chizat et al. (2019).

A.7. Expanding nonlinear networks

The first expansion (`scaled` in figures) rescales the input weights to each neuron by a constant factor α and the output weights the neuron by $1/\alpha$, which preserves the output of each neuron due to the scale-invariance of ReLUs. However, the magnitude of the representations that this model employs increases. The second expansion (`nuisance` in figures) adds nuisance neurons with random incoming weights and zero outgoing weights ("zero-type" neurons per Martinelli et al. 2023); similarly, these neurons do not affect the output the model, but introduce noise in hidden-layer reprsentations. Lastly, we include a parameter-expanded baseline (`duplicated`) that duplicates each neuron and correspondingly recales outgoing weights from the duplicated neuron and its copy by a factor of two; this manipulation does not change the representational structure of the model.

A.8. Figure 5

The neural networks start with $N_i = 784$, $N_h = 1024$, $N_o = 10$. Manipulations adding parameters in panel (E) add twice the number of neurons. Models are trained for 30 epochs on the full MNIST training set of 50,000 images with exponential learning rate decay.

B. Notation and preliminaries

Throughout this paper we adhere to the following notation: Scalars are denoted by letters (e.g. a, τ, L), matrices by bold uppercase letters (e.g. $\mathbf{X}$, $\mathbf{\Gamma}$), column vectors are denoted by bold lowercase letters (e.g. $\mathbf{x}$, $\boldsymbol{\lambda}$), and row vectors by the transpose of a column vector (e.g. $\mathbf{x}^T$, $\boldsymbol{\lambda}^T$). The vector dot product and vector outer product are denoted by $\mathbf{a}^T\mathbf{b}$ and $\mathbf{a}\mathbf{b}^T$ respectively. The zero matrix and identity matrix of size n are denoted by $\mathbf{0}$ and $\mathbf{I}_n$ respectively. $\mathbf{A}^{-1}$ and $\mathbf{A}^+$ denote the inverse and Moore–Penrose pseudoinverse of a matrix. The ℓ_2-norm of a vector is expressed by $||\mathbf{x}||_2$ and the Frobenius norm of a matrix is expressed by $||\mathbf{A}||_F$. Finally, the expected value and trace operator are denoted by $\langle\cdot\rangle$ and $\mathrm{Tr}(\cdot)$.

B.1. Compact singular value decomposition

We extensively use the cSVD to analyse matrix structures. For any matrix $\mathbf{A} \in \mathcal{R}^{m\times n}$ with rank r, the cSVD decomposes it into a product of three:

$$\mathrm{cSVD}(\mathbf{A}) = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{45}$$

where $\mathbf{U} \in \mathcal{R}^{m\times r}$ and $\mathbf{V} \in \mathcal{R}^{n\times r}$ are (semi-)orthonormal matrices containing the left and right singular vectors, and $\mathbf{S} \in \mathcal{R}^{r\times r}$ is a diagonal matrix with corresponding non-zero singular values in descending order. In contrast to the full singular value decomposition, the singular vectors of the cSVD are not generally square orthogonal matrices but may be semi-orthogonal dependent on the relationship of m, n, and r (Table 1). It further follows, that $\mathbf{S}^{-1}$ is well defined. Using that $(\mathbf{B}\mathbf{C})^+ = \mathbf{C}^+\mathbf{B}^+$ if $\mathbf{B}$ has orthonormal columns or $\mathbf{C}$ has orthonormal rows (Greville 1966) it further follows that

$$\begin{aligned}\mathbf{A}^+ &= \left(\mathbf{U}\mathbf{S}\mathbf{V}^T\right)^+ \\ &= \mathbf{V}\left(\mathbf{U}\mathbf{S}\right)^+ \\ &= \mathbf{V}\mathbf{S}^+\mathbf{U}^T \\ &= \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\end{aligned} \tag{46}$$

is the Moore-Penrose inverse of any matrix $\mathbf{A}$.

Table 1: Orthonormality of singular vectors of the cSVD

	$m = n$		$m < n$		$m > n$	
	$r = m$	$r < m$	$r = m$	$r < m$	$r = n$	$r < n$
$\mathbf{U}^T\mathbf{U}$	$= \mathbf{I}_m$	$= \mathbf{I}_r$	$= \mathbf{I}_m$	$= \mathbf{I}_r$	$= \mathbf{I}_n$	$= \mathbf{I}_r$
$\mathbf{U}\mathbf{U}^T$	$= \mathbf{I}_m$	$\neq \mathbf{I}_m$	$= \mathbf{I}_m$	$\neq \mathbf{I}_m$	$= \mathbf{I}_m$	$\neq \mathbf{I}_m$
$\mathbf{V}^T\mathbf{V}$	$= \mathbf{I}_m$	$= \mathbf{I}_r$	$= \mathbf{I}_m$	$= \mathbf{I}_r$	$= \mathbf{I}_n$	$= \mathbf{I}_r$
$\mathbf{V}\mathbf{V}^T$	$= \mathbf{I}_m$	$\neq \mathbf{I}_m$	$\neq \mathbf{I}_n$	$\neq \mathbf{I}_n$	$= \mathbf{I}_n$	$\neq \mathbf{I}_n$

B.2. Linearity of expected value and trace

Both, the expected value and trace operator are linear and therefore additive

$$\langle\mathbf{A} + \mathbf{B}\rangle = \langle\mathbf{A}\rangle + \langle\mathbf{B}\rangle, \tag{47}$$

$$\mathrm{Tr}\left(\mathbf{A} + \mathbf{B}\right) = \mathrm{Tr}(\mathbf{A}) + \mathrm{Tr}(\mathbf{B}), \tag{48}$$

and homogeneous

$$\langle a\mathbf{A}\rangle = a\langle\mathbf{A}\rangle, \tag{49}$$

$$\mathrm{Tr}(a\mathbf{A}) = a\,\mathrm{Tr}(\mathbf{A}), \tag{50}$$

from which it further follows that

$$\langle\mathrm{Tr}(\mathbf{A})\rangle = \mathrm{Tr}(\langle\mathbf{A}\rangle). \tag{51}$$

B.3. Expected values of random vectors and matrices

Let $\boldsymbol{\xi}_1 \in \mathcal{R}^n$ be a vector whose entries are i.i.d. and drawn from a zero-centred random distribution with variance σ_1^2. Then,

$$\langle \boldsymbol{\xi}_1 \rangle = \mathbf{0} \in \mathcal{R}^n \tag{52}$$

and the expected value of the inner and outer product is

$$\langle \boldsymbol{\xi}_1^T \boldsymbol{\xi}_1 \rangle = n\sigma_1^2 \quad \text{and} \quad \langle \boldsymbol{\xi}_1 \boldsymbol{\xi}_1^T \rangle = \sigma_1^2 \mathbf{I}_n \tag{53}$$

respectively. Let $\boldsymbol{\xi}_2 \in \mathcal{R}^m$ be a second random i.i.d. vector sampled from a zero-centred distribution with variance σ_2^2, then

$$\langle \boldsymbol{\xi}_1^T \boldsymbol{\xi}_2 \rangle = 0 \quad \text{and} \quad \langle \boldsymbol{\xi}_1 \boldsymbol{\xi}_2^T \rangle = \mathbf{0} \in \mathcal{R}^{m \times n}. \tag{54}$$

We continue by deriving general forms for the expected value of products of random matrices. Let $\boldsymbol{\Xi}_1 \in \mathcal{R}^{m \times n}$ be a matrix whose entries are i.i.d., drawn from a zero-centred random distribution with variance σ_1^2. Then,

$$\langle \boldsymbol{\Xi}_1 \rangle = \mathbf{0} \in \mathcal{R}^{m \times n}. \tag{55}$$

Further, let $\boldsymbol{\Xi}_{1i}$ and $\boldsymbol{\Xi}_{1j}$ denote the i-th and j-th column of the matrix. Then it follows from Equation (53) that the expected value of the inner product for each column of $\boldsymbol{\Xi}_1$ is

$$\left\langle \boldsymbol{\Xi}_{1i}^T \boldsymbol{\Xi}_{1j} \right\rangle = \begin{cases} 0, & \text{if } i \neq j \\ m\sigma_1^2, & \text{otherwise} \end{cases}, \tag{56}$$

and therefore that

$$\left\langle \boldsymbol{\Xi}_1^T \boldsymbol{\Xi}_1 \right\rangle = m\sigma_1^2 \mathbf{I}_n. \tag{57}$$

Similarly, from Equation (53) it follow that

$$\left\langle \boldsymbol{\Xi}_{1i} \boldsymbol{\Xi}_{1j}^T \right\rangle = \begin{cases} \mathbf{0}, & \text{if } i \neq j \\ \sigma_1^2 \mathbf{I}, & \text{otherwise} \end{cases} \tag{58}$$

and therefore that

$$\left\langle \boldsymbol{\Xi}_1 \boldsymbol{\Xi}_1^T \right\rangle = n\sigma_1^2 \mathbf{I}_m. \tag{59}$$

Let $\boldsymbol{\Xi}_2 \in \mathbb{R}^{l \times n}$ be a second random i.i.d. matrix, sampled from a zero-centred distribution with variance σ_2^2, then it follows from Appendix B.3 that

$$\left\langle \boldsymbol{\Xi}_2 \boldsymbol{\Xi}_1^T \right\rangle = \mathbf{0}. \tag{60}$$

We proceed by deriving equalities for the expected values of random matrices and their interactions with arbitrary constant matrices. For arbitrary constant matrix $\mathbf{B} \in \mathcal{R}^{m \times m}$ and $i \neq j$ we get

$$\begin{aligned} \left\langle \boldsymbol{\Xi}_1^T \mathbf{B} \boldsymbol{\Xi}_1 \right\rangle_{i,j} &= \operatorname{Tr}\left(\mathbf{B} \left\langle \boldsymbol{\Xi}_{1_j} \boldsymbol{\Xi}_{1_i}^T \right\rangle\right) \\ &= \operatorname{Tr}\left(\mathbf{B}\mathbf{0}\right) \\ &= 0 \end{aligned} \tag{61}$$

and for $i = j$

$$\begin{aligned} \left\langle \boldsymbol{\Xi}_1^T \mathbf{B} \boldsymbol{\Xi}_1 \right\rangle_{i,j} &= \operatorname{Tr}\left(\mathbf{B} \left\langle \boldsymbol{\Xi}_{1_j} \boldsymbol{\Xi}_{1_i}^T \right\rangle\right) \\ &= \operatorname{Tr}\left(\mathbf{B}\sigma_1^2 \mathbf{I}\right) \\ &= \sigma_1^2 \operatorname{Tr}\left(\mathbf{B}\right) \end{aligned} \tag{62}$$

and therefore

$$\left\langle \boldsymbol{\Xi}_1^T \mathbf{B} \boldsymbol{\Xi}_1 \right\rangle = \sigma_1^2 \operatorname{Tr}\left(\mathbf{B}\right) \mathbf{I}. \tag{63}$$

C. The general linear solution

From Theorem 3 in Laurent and Brecht (2018) it follows that any minimum of the convex and differentiable mean-squared error

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2P}\sum_{n=1}^{P} ||\mathbf{W}_2\mathbf{W}_1\mathbf{x}_n - \mathbf{y}_n||_2^2 \tag{64}$$

corresponds to the global optimum of the convex single-layer optimisation problem

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2P}\sum_{n=1}^{P} ||\bar{\mathbf{W}}\mathbf{x}_n - \mathbf{y}||_2^2, \tag{65}$$

as long as the underlying network

$$\bar{\mathbf{W}} = \mathbf{\Omega}_2\mathbf{\Omega}_1, \tag{66}$$

has no bottlenecks. The global optima of the network then correspond to

$$\begin{aligned}
& & \frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \bar{\mathbf{W}}} &= 0 \\
&\Leftrightarrow & \frac{1}{P}\sum_{n=1}^{P}\left(\bar{\mathbf{W}}\mathbf{x}_n - \mathbf{y}_n\right)\mathbf{x}_n^T &= 0 \\
&\Leftrightarrow & \bar{\mathbf{W}}\frac{1}{P}\sum_{n=1}^{P}\mathbf{x}_n\mathbf{x}_n^T - \frac{1}{P}\sum_{n=1}^{P}\mathbf{y}_n\mathbf{x}_n^T &= 0 \\
&\Leftrightarrow & \bar{\mathbf{W}}\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx}.
\end{aligned} \tag{67}$$

Resubstitution then gives the general linear solution

$$\mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx}. \tag{68}$$

Finally, we note that the GLS can also be written as

$$\begin{aligned}
& & \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx} \\
&\Leftrightarrow & \mathbf{\Omega}_2\mathbf{\Omega}_1 &= \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \mathbf{Z}(\mathbf{I} - \mathbf{\Sigma}_{xx}\mathbf{\Sigma}_{xx}^{+}) \\
&\Leftrightarrow & \mathbf{\Omega}_2\mathbf{\Omega}_1 &= \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \mathbf{Z}(\mathbf{I} - 1/P\mathbf{A}\mathbf{B}^2\mathbf{A}^T P\mathbf{A}\mathbf{B}^{-2}\mathbf{A}^T) \\
&\Leftrightarrow & \mathbf{\Omega}_2\mathbf{\Omega}_1 &= \mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{Z}(\mathbf{I} - \mathbf{A}\mathbf{A}^T),
\end{aligned} \tag{69}$$

where $\mathbf{Z} \in \mathcal{R}^{N_o \times N_i}$ is an arbitrary matrix.

Proof of Theorem 3.1. In the following, we derive a complete parametrisation of the GLS

$$\mathbf{\Omega}_2\mathbf{\Omega}_1 = \mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{Z}(\mathbf{I} - \mathbf{A}\mathbf{A}^T). \tag{70}$$

We begin by rewriting $\mathbf{\Omega}_1$ in the basis of relevant, irrelevant and unobserved null directions

$$\mathbf{\Omega}_1 = \mathbf{\Omega}_1\mathbf{P}_{\text{r}} + \mathbf{\Omega}_1\mathbf{P}_{\text{i}} + \mathbf{\Omega}_1\mathbf{P}_{\text{u}} \tag{71}$$

with corresponding projectors $\mathbf{P}_{\text{r}} = \mathbf{V}\mathbf{V}^T$, $\mathbf{P}_{\text{i}} = \mathbf{A}\mathbf{A}^T - \mathbf{V}\mathbf{V}^T$, and $\mathbf{P}_{\text{u}} = \mathbf{I} - \mathbf{A}\mathbf{A}^T$. Substitution into the GLS (Equation (69)) than reveals that directions that lie in the relevant input space must obey

$$\begin{aligned}
& & \mathbf{\Omega}_2(\mathbf{\Omega}_1\mathbf{P}_{\text{r}} + \mathbf{\Omega}_1\mathbf{P}_{\text{i}} + \mathbf{\Omega}_1\mathbf{P}_{\text{u}})\mathbf{P}_{\text{r}} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{Z}\mathbf{P}_{\text{u}})\mathbf{P}_{\text{r}} \\
&\Leftrightarrow & \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{P}_{\text{r}} &= \mathbf{U}\mathbf{S}\mathbf{V}^T,
\end{aligned} \tag{72}$$

from which it follows that $\mathbf{\Omega}_1\mathbf{P}_\mathrm{r} = \mathbf{\Omega}_1\mathbf{V}\mathbf{V}^T$ can be parametrised as $\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T$, where $\mathbf{Q} \in \mathcal{R}^{N_h \times r}$ is any full-column-rank matrix, projecting relevant inputs into hidden space. It then follows that we can further separate $\mathbf{\Omega}_2\mathbf{P}_\mathrm{h}$ into two subspaces, one projecting relevant hidden representations and one projecting all other dimensions

$$\begin{aligned} && \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{P}_\mathrm{r} &= \mathbf{U}\mathbf{S}\mathbf{V}^T \\ \Leftrightarrow && (\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+))\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T &= \mathbf{U}\mathbf{S}\mathbf{V}^T \\ \Leftrightarrow && \mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T &= \mathbf{U}\mathbf{S}\mathbf{V}^T \\ \Leftrightarrow && \mathbf{\Omega}_2\mathbf{Q} &= \mathbf{U}\sqrt{\mathbf{S}}, \end{aligned} \tag{73}$$

and thus $\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+$ can be parametrised as $\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+$. We call these two parts of the network function the core. We continue by analysing the subspace covered by irrelevant inputs

$$\begin{aligned} && (\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+))(\mathbf{\Omega}_1\mathbf{P}_\mathrm{r} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{i} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{u})\mathbf{P}_\mathrm{i} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{Z}\mathbf{P}_\mathrm{n})\mathbf{P}_\mathrm{i} \\ \Leftrightarrow && (\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+))\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} &= \mathbf{0} \\ \Leftrightarrow && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}. \end{aligned} \tag{74}$$

This implies that task-irrelevant inputs can be mapped into the core readout space by input weights (right-hand side of the equation), provided that other task-irrelevant components are simultaneously projected in such a way that they cancel the effect (left-hand side of the equation). This ensure that the network output remains unchanged despite the first-layer weights processing task-irrelevant components. For the compensation in the two-layer network to be successful, we have to have

$$\begin{aligned} && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} \\ \Leftrightarrow && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i} \\ \Leftrightarrow && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\right]^+ \\ && &\quad + \tilde{\mathbf{Z}}\left[\mathbf{I} - (\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i})^+\right](\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) \\ \Leftrightarrow && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\right]^+ \\ && &\quad + \tilde{\mathbf{Z}}\left[\mathbf{Q}\mathbf{Q}^+ + (\mathbf{I}-\mathbf{H}\mathbf{H}^+)\right](\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) \\ \Leftrightarrow && \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\right]^+ + \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+), \end{aligned} \tag{75}$$

where $\tilde{\mathbf{Z}} \in \mathcal{R}^{N_o \times N_h}$ is an arbitrary matrix. Crucially, the pseudo-inverse only exists if

$$\mathrm{im}(\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}) \subseteq \mathrm{im}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}). \tag{76}$$

As both sides live in the same subspace $\mathbf{P}_\mathrm{i}$, this is equivalent to

$$\mathrm{rank}(\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}). \tag{77}$$

Or in words, there must be at least as many dimensions of the irrelevant input space that are projected outside the core as there are dimensions that are projected into the core. Substitution then gives

$$\mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) = -\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\right]^+ + \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+). \tag{78}$$

Finally, we analyse the unoccupied null directions

$$\begin{aligned} && \mathbf{\Omega}_2(\mathbf{\Omega}_1\mathbf{P}_\mathrm{r} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{i} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{u})\mathbf{P}_\mathrm{u} &= (\mathbf{U}\mathbf{S}\mathbf{V}^T + \mathbf{Z}\mathbf{P}_\mathrm{u})\mathbf{P}_\mathrm{u} \\ \Leftrightarrow && \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{P}_\mathrm{u} &= \mathbf{Z}\mathbf{P}_\mathrm{u}, \end{aligned} \tag{79}$$

from which it follows that $\mathbf{\Omega}_1\mathbf{P}_\mathrm{u}$ can be chosen arbitrarily. In summary, we then have

$$\begin{aligned} \mathbf{\Omega}_1 &= \mathbf{\Omega}_1\mathbf{P}_\mathrm{r} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{i} + \mathbf{\Omega}_1\mathbf{P}_\mathrm{u} \\ &= \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{\Gamma}_1\mathbf{P}_\mathrm{i} + \mathbf{\Gamma}_2\mathbf{P}_\mathrm{u} \end{aligned} \tag{80}$$

where $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2 \in \mathcal{R}^{N_h \times N_i}$ can be chose freely up to the constraint that $\mathrm{rank}(\mathbf{Q}\mathbf{Q}^+\mathbf{\Gamma}_1\mathbf{P}_\mathrm{i}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Gamma}_1\mathbf{P}_\mathrm{i})$, and

$$\begin{aligned} \mathbf{\Omega}_2 &= \mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \mathbf{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) \\ &= \mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \mathbf{\Psi} + \mathbf{\Gamma}_3(\mathbf{I}-\mathbf{H}\mathbf{H}^+), \end{aligned} \tag{81}$$

where $\mathbf{\Gamma}_3 \in \mathcal{R}^{N_o \times N_h}$ is an arbitrary matrix and $\mathbf{\Psi} = -\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_\mathrm{i}\right]^+$. □

D. Partitioning of the solution manifold

Proof of Theorem 3.3. We use the method of Lagrange multipliers to minimise

$$\underset{\mathbf{W}_1,\mathbf{W}_2}{\operatorname{argmin}} ||\mathbf{W}_2\mathbf{W}_1||_F^2 \tag{82}$$

under the constraint that

$$\mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx}. \tag{83}$$

We begin by substituting $\bar{\mathbf{\Omega}} = \mathbf{W}_2\mathbf{W}_1$. Then the Lagrangian is

$$\mathcal{L} = ||\bar{\mathbf{\Omega}}||_F^2 + \operatorname{Tr}\left(\mathbf{\Lambda}^T\left(\bar{\mathbf{\Omega}}\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx}\right)\right) \tag{84}$$

with gradients

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\bar{\mathbf{\Omega}}} &= 2\bar{\mathbf{\Omega}} + \mathbf{\Lambda}\mathbf{\Sigma}_{xx} = 0 \\ \Leftrightarrow \quad \bar{\mathbf{\Omega}} &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}, \end{aligned} \tag{85}$$

and

$$\frac{\partial\mathcal{L}}{\partial\mathbf{\Lambda}} = \bar{\mathbf{\Omega}}\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx} = 0. \tag{86}$$

Then starting from Equation (85) we have

$$\begin{aligned} \bar{\mathbf{\Omega}} &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx} \\ \Leftrightarrow \quad \bar{\mathbf{\Omega}}\mathbf{\Sigma}_{xx} &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{\Sigma}_{xx} \\ \Leftrightarrow \quad \mathbf{\Sigma}_{yx} &= -\frac{1}{2}\mathbf{\Lambda}\frac{1}{P}\mathbf{A}\mathbf{B}^2\mathbf{A}^T\frac{1}{P}\mathbf{A}\mathbf{B}^2\mathbf{A}^T \\ \Leftrightarrow \quad \mathbf{\Sigma}_{yx}P\mathbf{A}\mathbf{B}^{-2}\mathbf{A}^T &= -\frac{1}{2}\mathbf{\Lambda}\frac{1}{P}\mathbf{A}\mathbf{B}^2\mathbf{A}^T \\ \Leftrightarrow \quad \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+ &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}, \end{aligned} \tag{87}$$

where in the third step we substituted Equation (86). Then, resubstitution into Equation (85) yields

$$\bar{\mathbf{\Omega}} = \mathbf{\Omega}_2\mathbf{\Omega}_1 = \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+ = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{88}$$

Next, we derive a complete parametrisation of the LSS

$$\mathbf{\Omega}_2\mathbf{\Omega}_1 = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{89}$$

Again, as in the parametrisation for the GLS (Theorem 3.1), we rewrite $\mathbf{\Omega}_1$ in the basis of relevant, irrelevant and unobserved null directions

$$\mathbf{\Omega}_1 = \mathbf{\Omega}_1\mathbf{P}_{\mathrm{r}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{i}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{u}} \tag{90}$$

with corresponding projectors $\mathbf{P}_{\mathrm{r}} = \mathbf{V}\mathbf{V}^T$, $\mathbf{P}_{\mathrm{i}} = \mathbf{A}\mathbf{A}^T - \mathbf{V}\mathbf{V}^T$, and $\mathbf{P}_{\mathrm{u}} = \mathbf{I} - \mathbf{A}\mathbf{A}^T$. First, we note that the equations for the relevant and irrelevant input directions are identical to the ones presented in Theorem 3.1, that is

$$\begin{aligned} \mathbf{\Omega}_2(\mathbf{\Omega}_1\mathbf{P}_{\mathrm{r}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{i}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{u}})\mathbf{P}_{\mathrm{r}} &= \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{P}_{\mathrm{r}} \\ \Leftrightarrow \quad \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{P}_{\mathrm{r}} &= \mathbf{U}\mathbf{S}\mathbf{V}^T, \end{aligned} \tag{91}$$

and

$$\begin{aligned} (\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \mathbf{\Omega}_2(\mathbf{I} - \mathbf{Q}\mathbf{Q}^+))(\mathbf{\Omega}_1\mathbf{P}_{\mathrm{r}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{i}} + \mathbf{\Omega}_1\mathbf{P}_{\mathrm{u}})\mathbf{P}_{\mathrm{i}} &= \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{P}_{\mathrm{i}} \\ \Leftrightarrow \quad \mathbf{\Omega}_2(\mathbf{I} - \mathbf{Q}\mathbf{Q}^+)\mathbf{\Omega}_1\mathbf{P}_{\mathrm{i}} &= -\mathbf{\Omega}_2\mathbf{Q}\mathbf{Q}^+\mathbf{\Omega}_1\mathbf{P}_{\mathrm{i}}. \end{aligned} \tag{92}$$

From which it follows that

$$\begin{aligned} \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{r}} &= \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T && (93)\\ \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{i}} &= \boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}} && (94)\\ \boldsymbol{\Omega}_2\mathbf{Q}\mathbf{Q}^+ &= \mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ && (95)\\ \boldsymbol{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+) &= \boldsymbol{\Psi} + \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+), && (96) \end{aligned}$$

where $\mathbf{Q} \in \mathcal{R}^{N_h \times r}$ is an arbitrary full-column-rank matrix, $\boldsymbol{\Gamma}_1$, is an arbitrary matrix subject to the constraint $\mathrm{rank}(\mathbf{Q}\mathbf{Q}^+\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}})$, $\boldsymbol{\Psi} = -\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}}\right]^+$, and $\tilde{\mathbf{Z}} \in \mathcal{R}^{N_o \times N_h}$ is an arbitrary matrix. However, LSS differ in how they constrain the solution manifold in the case of unobserved null directions. In particular, $\hat{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+)$ is constrained by

$$\begin{aligned} & \boldsymbol{\Omega}_2(\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{r}} + \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{i}} + \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}})\mathbf{P}_{\mathrm{u}} = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{P}_{\mathrm{u}}\\ \Leftrightarrow\quad & (\boldsymbol{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \boldsymbol{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+))\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}} = \mathbf{0}\\ \Leftrightarrow\quad & (\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi} + \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+))\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}} = \mathbf{0}\\ \Leftrightarrow\quad & \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+)(\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}} = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi})\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\\ \Leftrightarrow\quad & \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+) = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi})\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\left[(\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\right]^+ \qquad (97)\\ & \qquad + \hat{\mathbf{Z}}[\mathbf{I} - (\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}((\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}})^+](\mathbf{I}-\mathbf{H}\mathbf{H}^+)\\ \Leftrightarrow\quad & \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+) = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi})\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\left[(\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\right]^+\\ & \qquad + \hat{\mathbf{Z}}[\mathbf{H}\mathbf{H}^+ + (\mathbf{I}-\boldsymbol{\Omega}_1\boldsymbol{\Omega}_1^+)](\mathbf{I}-\mathbf{H}\mathbf{H}^+)\\ \Leftrightarrow\quad & \tilde{\mathbf{Z}}(\mathbf{I}-\mathbf{H}\mathbf{H}^+) = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi})\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\left[(\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\right]^+ + \hat{\mathbf{Z}}(\mathbf{I}-\boldsymbol{\Omega}_1\boldsymbol{\Omega}_1^+), \end{aligned}$$

where $\hat{\mathbf{Z}} \in \mathcal{R}^{N_o \times N_h}$ is an arbitrary matrix. Again, for the pseudo-inverse to exist we must have

$$\mathrm{im}(\mathbf{H}\mathbf{H}^+\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}) \subseteq \mathrm{im}((\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}). \qquad (98)$$

As both sides live in the same subspace $\mathbf{P}_{\mathrm{u}}$, this is equivalent to

$$\mathrm{rank}(\mathbf{H}\mathbf{H}^+\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}). \qquad (99)$$

In other words, at least as many unobserved dimensions that are projected into the occupied hidden-layer dimensions have to be projected into the unoccupied hidden-layer dimensions in order for a correction to exist. In summary, we then have

$$\begin{aligned} \boldsymbol{\Omega}_1 &= \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{r}} + \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{i}} + \boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\\ &= \mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}} + \boldsymbol{\Gamma}_2\mathbf{P}_{\mathrm{u}} \end{aligned} \qquad (100)$$

where $\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2 \in \mathcal{R}^{N_h \times N_i}$ can be chose freely up to the constraints that $\mathrm{rank}(\mathbf{Q}\mathbf{Q}^+\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\boldsymbol{\Gamma}_1\mathbf{P}_{\mathrm{i}})$, and $\mathrm{rank}(\mathbf{H}\mathbf{H}^+\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}) \leq \mathrm{rank}((\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}})$, and

$$\begin{aligned} \boldsymbol{\Omega}_2 &= \boldsymbol{\Omega}_2\mathbf{Q}\mathbf{Q}^+ + \boldsymbol{\Omega}_2(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\\ &= \mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi} + \boldsymbol{\Phi} + \boldsymbol{\Gamma}_3(\mathbf{I}-\boldsymbol{\Omega}_1\boldsymbol{\Omega}_1^+), \end{aligned} \qquad (101)$$

where $\boldsymbol{\Gamma}_3 \in \mathcal{R}^{N_o \times N_h}$ is an arbitrary matrix, $\boldsymbol{\Psi} = -\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{i}}\left[(\mathbf{I}-\mathbf{Q}\mathbf{Q}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{i}}\right]^+$, and $\boldsymbol{\Phi} = -(\mathbf{U}\sqrt{\mathbf{S}}\mathbf{Q}^+ + \boldsymbol{\Psi})\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\left[(\mathbf{I}-\mathbf{H}\mathbf{H}^+)\boldsymbol{\Omega}_1\mathbf{P}_{\mathrm{u}}\right]^+$. □

Proof of Theorem 3.5. We use the method of Lagrange multipliers to minimise

$$\underset{\mathbf{W}_1,\mathbf{W}_2}{\mathrm{argmin}}\, ||\mathbf{W}_1\mathbf{X}||_F^2 + ||\mathbf{W}_2||_F^2 \qquad (102)$$

under the constraint that

$$\mathbf{W}_2\mathbf{W}_1\boldsymbol{\Sigma}_{xx} = \boldsymbol{\Sigma}_{yx}. \qquad (103)$$

Then the Lagrangian is

$$\mathcal{L} = \ ||\mathbf{W}_1\mathbf{X}||_F^2 + ||\mathbf{W}_2||_F^2 + \mathrm{Tr}\left(\mathbf{\Lambda}^T\left(\mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx}\right)\right) \tag{104}$$

with gradients

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_1} &= 2\mathbf{W}_1\mathbf{X}\mathbf{X}^T + \mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx} = 0 \\ \Leftrightarrow \quad \mathbf{W}_1\mathbf{X}\mathbf{X}^T &= -\frac{1}{2}\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx}, \end{aligned} \tag{105}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_2} &= 2\mathbf{W}_2 + \mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T = 0 \\ \Leftrightarrow \quad \mathbf{W}_2 &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T, \end{aligned} \tag{106}$$

and

$$\frac{\partial \mathcal{L}}{\partial \mathbf{\Lambda}} = \mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx} = 0. \tag{107}$$

Starting from Equation (106), we then have

$$\begin{aligned} \mathbf{W}_2 &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{W}_2^T\mathbf{W}_2 &= -\frac{1}{2}\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{W}_2^T\mathbf{W}_2 &= \mathbf{W}_1\mathbf{X}\mathbf{X}^T\mathbf{W}_1^T, \end{aligned} \tag{108}$$

where in the last line we substituted Equation (105). Let

$$\mathrm{cSVD}(\mathbf{W}_2) = \mathbf{A}\mathbf{B}\mathbf{C}^T \text{ and } \mathrm{cSVD}(\mathbf{W}_1\mathbf{X}) = \mathbf{D}\mathbf{E}\mathbf{F}^T \tag{109}$$

be the cSVD of the network weights, then

$$\begin{aligned} \mathbf{W}_2^T\mathbf{W}_2 &= \mathbf{W}_1\mathbf{X}\mathbf{X}^T\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{C}\mathbf{B}\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{C}^T &= \mathbf{D}\mathbf{E}\mathbf{F}^T\mathbf{F}\mathbf{E}\mathbf{D}^T \\ \Leftrightarrow \quad \mathbf{C}\mathbf{B}^2\mathbf{C}^T &= \mathbf{D}\mathbf{E}^2\mathbf{D}^T. \end{aligned} \tag{110}$$

Since $\mathbf{C}$ and $\mathbf{D}$ are (semi-)orthonormal matrices and $\mathbf{B}$ and $\mathbf{E}$ are diagonal matrices with strictly positive entries it follows that

$$\mathbf{C} = \mathbf{D} \;\text{ and }\; \mathbf{B}^2 = \mathbf{E}^2 \Leftrightarrow \mathbf{B} = \mathbf{E}. \tag{111}$$

In the following we denote $\mathbf{C}$ and $\mathbf{D}$ as $\mathbf{R}$, and $\mathbf{B}$ and $\mathbf{E}$ as $\mathbf{G}$ and write

$$\mathbf{W}_1\mathbf{X} = \mathbf{R}\mathbf{G}\mathbf{F}^T \text{ and } \mathbf{W}_2 = \mathbf{A}\mathbf{G}\mathbf{R}^T, \tag{112}$$

where $\mathbf{R}$ is an arbitrary (semi-)orthogonal matrix. Finally, let

$$\mathrm{cSVD}(\mathbf{X}) = \mathbf{J}\mathbf{K}\mathbf{L}^T, \tag{113}$$

and

$$\begin{aligned} \mathrm{cSVD}(\mathbf{Y}\mathbf{X}^T\mathbf{X}^{+T}) &= \mathrm{cSVD}(\mathbf{Y}\mathbf{L}\mathbf{K}\mathbf{J}^T\mathbf{J}\mathbf{K}^{-1}\mathbf{L}^T) \\ &= \mathrm{cSVD}(\mathbf{Y}\mathbf{L}\mathbf{L}^T) \\ &= \mathbf{M}\mathbf{N}\mathbf{O}^T. \end{aligned} \tag{114}$$

Then starting from Equation (107) we get

$$\begin{aligned} \mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx} \\ \Leftrightarrow \quad \mathbf{W}_2\mathbf{W}_1\mathbf{X}\mathbf{X}^T &= \mathbf{Y}\mathbf{X}^T \\ \Leftrightarrow \quad \mathbf{W}_2\mathbf{W}_1\mathbf{J}\mathbf{K}^2\mathbf{J}^T &= \mathbf{Y}\mathbf{L}\mathbf{K}\mathbf{J}^T \\ \Leftrightarrow \quad \mathbf{W}_2\mathbf{W}_1\mathbf{J}\mathbf{K}\mathbf{L}^T &= \mathbf{Y}\mathbf{L}\mathbf{L}^T \\ \Leftrightarrow \quad \mathbf{A}\mathbf{G}\mathbf{R}^T\mathbf{R}\mathbf{G}\mathbf{F}^T &= \mathbf{M}\mathbf{N}\mathbf{O}^T \\ \Leftrightarrow \quad \mathbf{A}\mathbf{G}^2\mathbf{F}^T &= \mathbf{M}\mathbf{N}\mathbf{O}^T, \end{aligned} \tag{115}$$

from which it follows that

$$\mathbf{A} = \mathbf{M}, \ \mathbf{F}^T = \mathbf{O}^T, \text{ and } \mathbf{G}^2 = \mathbf{N} \Leftrightarrow \mathbf{G} = \sqrt{\mathbf{N}} \tag{116}$$

and therefore that

$$\mathbf{W}_2 = \mathbf{M}\sqrt{\mathbf{N}}\mathbf{R}^T \text{ and } \mathbf{W}_1\mathbf{X} = \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T. \tag{117}$$

Finally, we rewrite

$$\begin{aligned} \mathbf{W}_1\mathbf{X} &= \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T \\ \mathbf{W}_1\mathbf{J}\mathbf{K}\mathbf{L}^T &= \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T \\ \mathbf{W}_1 &= \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{L}\mathbf{K}^{-1}\mathbf{J}^T + \mathbf{Z}\left(\mathbf{I} - \mathbf{J}\mathbf{J}^T\right) \\ \mathbf{W}_1 &= \mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+ + \underbrace{\mathbf{Z}\left(\mathbf{I} - \mathbf{J}\mathbf{J}^T\right)}_{\mathbf{\Gamma}}, \end{aligned} \tag{118}$$

where $\mathbf{Z}$ is an arbitrary matrix. □

Proof of Theorem 3.7. We use the method of Lagrange multipliers to minimise

$$\underset{\mathbf{W}_1,\mathbf{W}_2}{\operatorname{argmin}} ||\mathbf{W}_1||_F^2 + ||\mathbf{W}_2||_F^2 \tag{119}$$

under the constraint that

$$\mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} = \mathbf{\Sigma}_{yx}. \tag{120}$$

Then the Lagrangian is

$$\mathcal{L} = ||\mathbf{W}_1||_F^2 + ||\mathbf{W}_2||_F^2 + \operatorname{Tr}\left(\mathbf{\Lambda}^T\left(\mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx}\right)\right) \tag{121}$$

with gradients

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{W}_1} &= 2\mathbf{W}_1 + \mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx} = 0 \\ &\Leftrightarrow \mathbf{W}_1 = -\frac{1}{2}\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx}, \end{aligned} \tag{122}$$

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{W}_2} &= 2\mathbf{W}_2 + \mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T = 0 \\ &\Leftrightarrow \mathbf{W}_2 = -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T, \end{aligned} \tag{123}$$

and

$$\frac{\partial\mathcal{L}}{\partial\mathbf{\Lambda}} = \mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} - \mathbf{\Sigma}_{yx} = 0. \tag{124}$$

Starting from Equation (123) we then have

$$\begin{aligned} \mathbf{W}_2 &= -\frac{1}{2}\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{W}_2^T\mathbf{W}_2 &= -\frac{1}{2}\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{W}_2^T\mathbf{W}_2 &= \mathbf{W}_1\mathbf{W}_1^T, \end{aligned} \tag{125}$$

where in the last line we substituted Equation (122). Let

$$\operatorname{cSVD}(\mathbf{W}_2) = \mathbf{A}\mathbf{B}\mathbf{C}^T \text{ and } \operatorname{cSVD}(\mathbf{W}_1) = \mathbf{D}\mathbf{E}\mathbf{F}^T \tag{126}$$

be the cSVD of the network weights, then

$$\begin{aligned} \mathbf{W}_2^T\mathbf{W}_2 &= \mathbf{W}_1\mathbf{W}_1^T \\ \Leftrightarrow \quad \mathbf{C}\mathbf{B}\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{C}^T &= \mathbf{D}\mathbf{E}\mathbf{F}^T\mathbf{F}\mathbf{E}\mathbf{D}^T \\ \Leftrightarrow \quad \mathbf{C}\mathbf{B}^2\mathbf{C}^T &= \mathbf{D}\mathbf{E}^2\mathbf{D}^T. \end{aligned} \tag{127}$$

Since $\mathbf{C}$ and $\mathbf{D}$ are (semi-)orthonormal matrices and $\mathbf{B}$ and $\mathbf{E}$ are diagonal matrices with strictly positive entries it follows that

$$\mathbf{C} = \mathbf{D} \text{ and } \mathbf{B}^2 = \mathbf{E}^2 \Leftrightarrow \mathbf{B} = \mathbf{E}. \tag{128}$$

It further follows that $\mathbf{W}_1$ and $\mathbf{W}_2$ must be of identical rank. In the following we denote $\mathbf{C}$ and $\mathbf{D}$ as $\mathbf{R}$, and $\mathbf{B}$ and $\mathbf{E}$ as $\mathbf{G}$ and write

$$\mathbf{W}_1 = \mathbf{RGF}^T \text{ and } \mathbf{W}_2 = \mathbf{AGR}^T, \tag{129}$$

where $\mathbf{R}$ is an arbitrary (semi-)orthogonal matrix. Finally, let

$$\mathrm{cSVD}(\mathbf{\Sigma}_{xx}) = \mathbf{JKJ}^T \tag{130}$$

and

$$\mathrm{cSVD}(\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^+) = \mathbf{USV}^T \tag{131}$$

denote the cSVD of the input covariance matrix and a least-squares solution. Then, starting from Equation (124) and using Equation (122), we get

$$\begin{aligned}
& & \mathbf{W}_2\mathbf{W}_1\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx} \\
&\Leftrightarrow & -\frac{1}{2}\mathbf{W}_2\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx}\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx} \\
&\Leftrightarrow & -\frac{1}{2}\mathbf{W}_2\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{J}\mathbf{K}^2\mathbf{J}^T &= \mathbf{\Sigma}_{yx} \\
&\Leftrightarrow & -\frac{1}{2}\mathbf{W}_2\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{JKJ}^T &= \mathbf{\Sigma}_{yx}\mathbf{J}\mathbf{K}^{-1}\mathbf{J}^T \\
&\Leftrightarrow & -\frac{1}{2}\mathbf{W}_2\mathbf{W}_2^T\mathbf{\Lambda}\mathbf{\Sigma}_{xx} &= \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^+ \\
&\Leftrightarrow & \mathbf{W}_2\mathbf{W}_1 &= \mathbf{USV}^T \\
&\Leftrightarrow & \mathbf{AGR}^T\mathbf{RGF}^T &= \mathbf{USV}^T \\
&\Leftrightarrow & \mathbf{A}\mathbf{G}^2\mathbf{F}^T &= \mathbf{USV}^T,
\end{aligned} \tag{132}$$

from which it follows that

$$\mathbf{A} = \mathbf{U}, \mathbf{F}^T = \mathbf{V}^T \text{ and } \mathbf{G}^2 = \mathbf{S} \Leftrightarrow \mathbf{G} = \sqrt{\mathbf{S}} \tag{133}$$

and therefore that

$$\mathbf{W}_2 = \mathbf{U}\sqrt{\mathbf{S}}\mathbf{R}^T \text{ and } \mathbf{W}_1 = \mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T. \tag{134}$$

□

E. Hidden-layer representations

Proposition E.1. *Let* $\mathrm{cSVD}(\mathbf{X}) = \mathbf{ABC}^T$*, then GLS are identical to LSS when operating on inputs* $\mathbf{X}$ *as any GLS (Equation (69)) can be written as*

$$\begin{aligned}
& \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{X} = \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+\mathbf{X} + \mathbf{Z}(\mathbf{I} - \mathbf{AA}^T)\mathbf{X} \\
\Leftrightarrow\ & \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{X} = \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+\mathbf{X} + \mathbf{Z}(\mathbf{I} - \mathbf{AA}^T)\mathbf{ABC}^T \\
\Leftrightarrow\ & \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{X} = \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+\mathbf{X} + \mathbf{Z}(\mathbf{ABC}^T - \mathbf{ABC}^T) \\
\Leftrightarrow\ & \mathbf{\Omega}_2\mathbf{\Omega}_1\mathbf{X} = \mathbf{\Sigma}_{yx}(\mathbf{\Sigma}_{xx})^+\mathbf{X}.
\end{aligned} \tag{135}$$

As a consequence, hidden-layer representations of the training data in GLS and LSS can be analysed by means of studying the LSS only.

Proof of Corollary 3.9. Following Proposition E.1 and given the parametrisation of $\mathbf{\Omega}_1$ as derived in Theorems 3.1 and 3.3, we derive

$$\begin{aligned}
\mathrm{RSM} &= \mathbf{X}^T\mathbf{\Omega}_1^T\mathbf{\Omega}_1\mathbf{X} \\
&= \mathbf{X}^T(\mathbf{V}\sqrt{\mathbf{S}}\mathbf{Q}^T + \mathbf{P}_{\mathrm{i}}^T\mathbf{\Gamma}_1^T + \mathbf{P}_{\mathrm{u}}^T\mathbf{\Gamma}_2^T)(\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{\Gamma}_1\mathbf{P}_{\mathrm{i}} + \mathbf{\Gamma}_2\mathbf{P}_{\mathrm{u}})\mathbf{X} \\
&= \mathbf{X}^T(\mathbf{V}\sqrt{\mathbf{S}}\mathbf{Q}^T\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{V}\sqrt{\mathbf{S}}\mathbf{Q}^T\mathbf{\Gamma}_1\mathbf{P}_{\mathrm{i}} + \mathbf{P}_{\mathrm{i}}^T\mathbf{\Gamma}_1^T\mathbf{Q}\sqrt{\mathbf{S}}\mathbf{V}^T + \mathbf{P}_{\mathrm{i}}^T\mathbf{\Gamma}_1^T\mathbf{\Gamma}_1\mathbf{P}_{\mathrm{i}})\mathbf{X}
\end{aligned} \tag{136}$$

□

Proof of Corollary 3.11. Let

$$\text{cSVD}(\mathbf{X}) = \mathbf{A}\mathbf{B}\mathbf{C}^T, \tag{137}$$

then given the subset of $\mathbf{\Omega}_1$ as defined in Theorem 3.5 we derive

$$\begin{aligned} \mathbf{X}^T(\mathbf{X}^{+T}\mathbf{O}\sqrt{\mathbf{N}}\mathbf{R}^T + \mathbf{\Gamma}^T)(\mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+ + \mathbf{\Gamma})\mathbf{X} &= \mathbf{X}^T\mathbf{X}^{+T}\mathbf{O}\sqrt{\mathbf{N}}\mathbf{R}^T\mathbf{R}\sqrt{\mathbf{N}}\mathbf{O}^T\mathbf{X}^+\mathbf{X} \\ &= \mathbf{C}\mathbf{B}\mathbf{A}^T\mathbf{A}\mathbf{B}^{-1}\mathbf{C}^T\mathbf{O}\mathbf{N}\mathbf{O}^T\mathbf{C}\mathbf{B}^{-1}\mathbf{A}^T\mathbf{A}\mathbf{B}\mathbf{C}^T \\ &= \mathbf{C}\mathbf{C}^T\mathbf{O}\mathbf{N}\mathbf{O}^T\mathbf{C}\mathbf{C}^T \\ &= \mathbf{O}\mathbf{N}\mathbf{O}^T \,, \end{aligned} \tag{138}$$

where in the last step we used that

$$\mathbf{Y}\mathbf{X}^T\mathbf{X}^{+T} = \mathbf{Y}\mathbf{C}\mathbf{C}^T. \tag{139}$$

□

Proof of Corollary 3.12. Given the subset of $\mathbf{\Omega}_1$ as defined in Theorem 3.7 we derive

$$\mathbf{X}^T\mathbf{V}\sqrt{\mathbf{S}}\mathbf{R}^T\mathbf{R}\sqrt{\mathbf{S}}\mathbf{V}^T\mathbf{X} = \mathbf{X}^T\mathbf{V}\mathbf{S}\mathbf{V}^T\mathbf{X}\,. \tag{140}$$

□

F. Secondary error

Proof of Theorem 5.1. Let the secondary error be denoted by

$$\mathcal{G}_{\text{MSE}} = \frac{1}{2Q}\sum_{n=1}^{Q}||\mathbf{W}_2\mathbf{W}_1\tilde{\mathbf{x}}_n - \tilde{\mathbf{y}}_n||_2^2, \tag{141}$$

where input vectors $\tilde{\mathbf{x}}_n$ and corresponding target values $\tilde{\mathbf{y}}_n$ come from a secondary dataset $\tilde{\mathcal{D}} = \{(\tilde{\mathbf{x}}_n, \tilde{\mathbf{y}}_n)\}_{n=1}^{Q}$ with a total of Q input-output pairs. We want to find the point on the solution manifold of the primary task such that $\mathcal{G}_{\text{MSE}}$ is minimised. We therefore substitute the GLS (see Equation (69)) into the secondary error

$$\mathcal{G}_{\text{MSE}} = \frac{1}{2Q}\sum_{n=1}^{Q}\left|\left|\left(\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \mathbf{Z}\mathbf{P}_{\text{u}}\right)\tilde{\mathbf{x}}_n - \tilde{\mathbf{y}}_n\right|\right|_2^2, \tag{142}$$

where $\mathbf{Z} \in \mathcal{R}^{N_o \times N_i}$ is an arbitrary matrix and $\mathbf{P}_{\text{u}} = \mathbf{I} - \mathbf{A}\mathbf{A}^T$. Thus $\mathbf{Z}\mathbf{P}_{\text{u}}$, which describes all possible transformations of the network function that lie in the unoccupied input space of the primary task is the only degree of freedom. Thus, we have to find $\tilde{\mathbf{Z}}$, which minimises the error. To this end, we define

$$\text{cSVD}(\mathbf{P}_{\text{u}}\tilde{\mathbf{X}}) = \mathbf{D}\mathbf{E}\mathbf{F}^T, \tag{143}$$

and continue with

$$\begin{aligned} \frac{\partial\mathcal{G}_{\text{MSE}}}{\partial\tilde{\mathbf{Z}}} = \frac{1}{Q}\sum_{n=1}^{Q}\left(\left(\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \tilde{\mathbf{Z}}\mathbf{P}_{\text{u}}\right)\tilde{\mathbf{x}}_n - \tilde{\mathbf{y}}_n\right)\tilde{\mathbf{x}}_n^T\mathbf{P}_{\text{u}} &= \mathbf{0} \\ \Leftrightarrow \quad \left(\left(\mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+} + \tilde{\mathbf{Z}}\mathbf{P}_{\text{u}}\right)\tilde{\mathbf{\Sigma}}_{xx} - \tilde{\mathbf{\Sigma}}_{yx}\right)\mathbf{P}_{\text{u}} &= \mathbf{0} \\ \Leftrightarrow \quad \tilde{\mathbf{Z}}\mathbf{P}_{\text{u}}\tilde{\mathbf{\Sigma}}_{xx}\mathbf{P}_{\text{u}} &= \left(\tilde{\mathbf{\Sigma}}_{yx} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{\Sigma}}_{xx}\right)\mathbf{P}_{\text{u}} \\ \Leftrightarrow \quad \tilde{\mathbf{Z}}\mathbf{P}_{\text{u}}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{P}_{\text{u}} &= \left(\tilde{\mathbf{Y}} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{X}}\right)\tilde{\mathbf{X}}^T\mathbf{P}_{\text{u}} \\ \Leftrightarrow \quad \tilde{\mathbf{Z}}\mathbf{D}\mathbf{E}^2\mathbf{D}^T &= \left(\tilde{\mathbf{Y}} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{X}}\right)\mathbf{F}\mathbf{E}\mathbf{D}^T \\ \Leftrightarrow \quad \tilde{\mathbf{Z}} &= \left(\tilde{\mathbf{Y}} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{X}}\right)\mathbf{F}\mathbf{E}^{-1}\mathbf{D}^T + \tilde{\mathbf{\Gamma}}(\mathbf{I} - \mathbf{D}\mathbf{D}^T) \\ \Leftrightarrow \quad \tilde{\mathbf{Z}} &= \left(\tilde{\mathbf{Y}} - \mathbf{\Sigma}_{yx}\mathbf{\Sigma}_{xx}^{+}\tilde{\mathbf{X}}\right)\left(\mathbf{P}_{\text{u}}\tilde{\mathbf{X}}\right)^{+} + \tilde{\mathbf{\Gamma}}(\mathbf{I} - \mathbf{D}\mathbf{D}^T), \end{aligned} \tag{144}$$

where $\tilde{\mathbf{\Gamma}} \in \mathcal{R}^{N_o \times N_i}$ is an arbitrary matrix. Note that $\tilde{\mathbf{\Gamma}}(\mathbf{I} - \mathbf{D}\mathbf{D}^T)\mathbf{P}_{\text{u}}$ describes all possible transformations that lie in the null spaces of both $\mathbf{X}$ and $\tilde{\mathbf{X}}$, and therefore have no effect on the processing of either the primary or secondary data. □

G. Input and parameter noise

Proof of Theorem 5.6. Let $\boldsymbol{\xi}_{\mathbf{x}_n}$ denote vectors and $\boldsymbol{\Xi}_1$ and $\boldsymbol{\Xi}_2$ denote matrices whose entries are i.i.d. and drawn from a zero-centred random distribution with variance $\sigma^2_{\mathbf{x}}$, $\sigma^2_{\boldsymbol{\Omega}_1}$ and $\sigma^2_{\boldsymbol{\Omega}_2}$ respectively. Then, we can write the expected mean-squared error under additive input noise and parameter noise as

$$\begin{aligned}
&\left\langle \frac{1}{2P}\sum_{n=1}^{P} ||\left(\boldsymbol{\Omega}_2+\boldsymbol{\Xi}_2\right)\left(\boldsymbol{\Omega}_1+\boldsymbol{\Xi}_1\right)\left(\mathbf{x}_n+\boldsymbol{\xi}_{\mathbf{x}_n}\right)-\mathbf{y}_n||_2^2\right\rangle \\
&=\frac{1}{2P}\sum_{n=1}^{P}\left\langle ||\underbrace{\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n-\mathbf{y}_n}_{a}+\underbrace{\big(\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1+\mathbf{W}_2\boldsymbol{\Xi}_1+}_{b}\underbrace{\boldsymbol{\Xi}_2\mathbf{W}_1+\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\big)\boldsymbol{\xi}_{\mathbf{x}_n}+\big(\mathbf{W}_2\boldsymbol{\Xi}_1+\boldsymbol{\Xi}_2\mathbf{W}_1+}_{b}\underbrace{\boldsymbol{\Xi}_2\boldsymbol{\Xi}_2\big)\mathbf{x}_n}_{b}||_2^2\right\rangle \qquad (145)\\
&=\frac{1}{2P}\sum_{n=1}^{P}\mathbf{a}^T\mathbf{a}+\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\langle\mathbf{b}\rangle+\frac{1}{2P}\sum_{n=1}^{P}\langle\mathbf{b}^T\mathbf{b}\rangle\,.
\end{aligned}$$

In the following we solve each of the three summands independently. Using the definition of a linear solution (Definition 2.3), we derive

$$\begin{aligned}
&\frac{1}{2P}\sum_{n=1}^{P}\mathbf{a}^T\mathbf{a}\\
&=\frac{1}{2P}\sum_{n=1}^{P}\mathbf{x}_n^T\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n-\frac{1}{2P}\sum_{n=1}^{P}\mathbf{x}_n^T\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\mathbf{y}_n-\frac{1}{2P}\sum_{n=1}^{P}\mathbf{y}_n^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n+\frac{1}{2P}\sum_{n=1}^{P}\mathbf{y}_n^T\mathbf{y}_n \qquad (146)\\
&=\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\boldsymbol{\Sigma}_{xx})-\operatorname{Tr}(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Sigma}_{yx})+\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}_{yy})\\
&=-\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\boldsymbol{\Sigma}_{xx}(\boldsymbol{\Sigma}_{xx})^+\boldsymbol{\Sigma}_{xx}^T)+\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}_{yy})\\
&=-\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}_{yx}(\boldsymbol{\Sigma}_{xx})^+\boldsymbol{\Sigma}_{yx}^T)+\frac{1}{2}\operatorname{Tr}(\boldsymbol{\Sigma}_{yy})\\
&=c,
\end{aligned}$$

which is a noise-independent constant that only depends on the statistics of the training data. Next, using the assumption that the noise is zero-centred, we derive that

$$\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\langle\mathbf{b}\rangle=0 \qquad (147)$$

using that

$$\langle\mathbf{b}\rangle=\Big\langle\big(\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1+\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1+\boldsymbol{\Xi}_2\boldsymbol{\Omega}_1+\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\big)\langle\boldsymbol{\xi}_{\mathbf{x}_n}\rangle+\big(\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1+\boldsymbol{\Xi}_2\boldsymbol{\Omega}_1+\boldsymbol{\Xi}_2\boldsymbol{\Xi}_2\big)\mathbf{x}_n\Big\rangle=0\,. \qquad (148)$$

Which leaves us with the third term, which we can write as

$$\begin{aligned}
\frac{1}{2P}\sum_{n=1}^{P}\langle\mathbf{b}^T\mathbf{b}\rangle=\frac{1}{2P}\sum_{n=1}^{P}\Big(&\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T(...)\boldsymbol{\xi}_{\mathbf{x}_n}\rangle+ &(149)\\
&\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T(...)\mathbf{x}_n\rangle+\langle\mathbf{x}_n^T(...)\boldsymbol{\xi}_{\mathbf{x}_n}\rangle+\mathbf{x}_n^T\langle(...)\rangle\mathbf{x}_n\Big). &(150)
\end{aligned}$$

In the following, we solve each of the four summands independently. We begin with

$$\begin{aligned}\Bigg\langle \boldsymbol{\xi}_{\mathbf{x}_n}^T \Bigg(&\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\Big\langle\boldsymbol{\Xi}_1\Big\rangle + \\ &\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\Big\langle\boldsymbol{\Xi}_2\Big\rangle\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\Big\langle\boldsymbol{\Xi}_2\Big\rangle\Big\langle\boldsymbol{\Xi}_1\Big\rangle + \\ &\Big\langle\boldsymbol{\Xi}_1^T\Big\rangle\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1 + \Big\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\Big\rangle + \\ &\Big\langle\boldsymbol{\Xi}_1^T\Big\rangle\boldsymbol{\Omega}_2^T\Big\langle\boldsymbol{\Xi}_2\Big\rangle\boldsymbol{\Omega}_1 + \Big\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\Big\langle\boldsymbol{\Xi}_2\Big\rangle\boldsymbol{\Xi}_1\Big\rangle + \\ &\boldsymbol{\Omega}_1^T\Big\langle\boldsymbol{\Xi}_2^T\Big\rangle\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1^T\Big\langle\boldsymbol{\Xi}_2^T\Big\rangle\boldsymbol{\Omega}_2\Big\langle\boldsymbol{\Xi}_1\Big\rangle + \\ &\boldsymbol{\Omega}_1^T\Big\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\Big\rangle\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1^T\Big\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\Big\rangle\Big\langle\boldsymbol{\Xi}_1\Big\rangle + \\ &\Big\langle\boldsymbol{\Xi}_1^T\Big\rangle\Big\langle\boldsymbol{\Xi}_2^T\Big\rangle\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1 + \Big\langle\boldsymbol{\Xi}_1^T\Big\langle\boldsymbol{\Xi}_2^T\Big\rangle\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\Big\rangle + \\ &\Big\langle\boldsymbol{\Xi}_1^T\Big\rangle\Big\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\Big\rangle\boldsymbol{\Omega}_1 + \Big\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\Big\rangle\Bigg)\boldsymbol{\xi}_{\mathbf{x}_n}\Bigg\rangle,\end{aligned} \tag{151}$$

where we used that noise is i.i.d.. Using that noise is zero-centred, we note that summands 2-5, 7-10, and 12-15 resolve to 0, which leaves us with summands 1, 6, 11, and 16 which we solve using Equations (53), (57), (59) and (63), as

$$\begin{aligned}&\Big\langle \boldsymbol{\xi}_{\mathbf{x}_n}^T\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\boldsymbol{\xi}_{\mathbf{x}_n}\Big\rangle \\ &= \operatorname{Tr}\left(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}\boldsymbol{\xi}_{\mathbf{x}_n}^T\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2||\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1||_F^2,\end{aligned} \tag{152}$$

$$\begin{aligned}&\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\right\rangle\boldsymbol{\xi}_{\mathbf{x}_n}\right\rangle \\ &= \operatorname{Tr}\left(\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\right\rangle\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}\boldsymbol{\xi}_{\mathbf{x}_n}^T\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2\operatorname{Tr}\left(\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\left\langle\boldsymbol{\Xi}_1\boldsymbol{\Xi}_1^T\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2\sigma_1^2N_i||\boldsymbol{\Omega}_2||_F^2,\end{aligned} \tag{153}$$

$$\begin{aligned}&\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T\boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1\boldsymbol{\xi}_{\mathbf{x}_n}\right\rangle \\ &= \operatorname{Tr}\left(\boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}\boldsymbol{\xi}_{\mathbf{x}_n}^T\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2\operatorname{Tr}\left(\boldsymbol{\Omega}_1\boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2\sigma_2^2N_o||\boldsymbol{\Omega}_1||_F^2,\end{aligned} \tag{154}$$

and

$$\begin{aligned}&\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T\boldsymbol{\Xi}_1^T\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\boldsymbol{\xi}_{\mathbf{x}_n}\right\rangle \\ &= \operatorname{Tr}\left(\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\right\rangle\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}\boldsymbol{\xi}_{\mathbf{x}_n}^T\right\rangle\right) \\ &= \sigma_{\mathbf{x}}^2\operatorname{Tr}\left(\left\langle\boldsymbol{\Xi}_1\boldsymbol{\Xi}_1^T\right\rangle\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\right) \\ &\quad\sigma_{\mathbf{x}}^2\sigma_1^2N_i\sigma_2^2N_o\operatorname{Tr}\left(\mathbf{I}_{N_h}\right) \\ &\quad\sigma_{\mathbf{x}}^2\sigma_1^2N_i\sigma_2^2N_oN_h.\end{aligned} \tag{155}$$

We continue by noting that

$$\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T(...)\mathbf{x}_n\rangle = \operatorname{Tr}\left(\langle(...)\rangle\mathbf{x}_n\langle\boldsymbol{\xi}_{\mathbf{x}_n}^T\rangle\right) = 0 \tag{156}$$

and

$$\left\langle \mathbf{x}_n^T(...)\boldsymbol{\xi}_{\mathbf{x}_n}\right\rangle = \mathrm{Tr}\left(\left\langle(...)\right\rangle\left\langle\boldsymbol{\xi}_{\mathbf{x}_n}\right\rangle\mathbf{x}_n^T\right) = 0. \tag{157}$$

Finally, we solve

$$\begin{aligned}\mathbf{x}_n^T\Bigg(&\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\right\rangle + \left\langle\boldsymbol{\Xi}_1^T\right\rangle\boldsymbol{\Omega}_2^T\left\langle\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1 +\\ &\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\left\langle\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Xi}_1\right\rangle + \boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\right\rangle\boldsymbol{\Omega}_2\left\langle\boldsymbol{\Xi}_1\right\rangle +\\ &\boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\left\langle\boldsymbol{\Xi}_1\right\rangle +\\ &\left\langle\boldsymbol{\Xi}_1^T\left\langle\boldsymbol{\Xi}_2^T\right\rangle\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\right\rangle + \left\langle\boldsymbol{\Xi}_1^T\right\rangle\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1 +\\ &\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\right\rangle\Bigg)\mathbf{x}_n,\end{aligned} \tag{158}$$

where we again used that that noise is i.i.d.. Using that noise is zero-centred, we note that summands 2-4, and 6-8 are equal to 0. Which leaves us with summands 1, 5, and 9, which we solve using Equations (57) and (63) as

$$\begin{aligned}&\mathbf{x}_n^T\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\right\rangle\mathbf{x}_n\\ &= \sigma_1^2\,\mathrm{Tr}(\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2)\mathbf{x}_n^T\mathbf{I}\mathbf{x}_n\\ &= \sigma_1^2||\boldsymbol{\Omega}_2||_F^2\,\mathrm{Tr}(\mathbf{x}_n\mathbf{x}_n^T),\end{aligned} \tag{159}$$

$$\begin{aligned}&\mathbf{x}_n^T\boldsymbol{\Omega}_1^T\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1\mathbf{x}\\ &= N_o\sigma_2^2\,\mathrm{Tr}(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_1\mathbf{x}_n\mathbf{x}_n^T),\end{aligned} \tag{160}$$

and

$$\begin{aligned}&\mathbf{x}_n^T\left\langle\boldsymbol{\Xi}_1^T\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\right\rangle\mathbf{x}\\ &= \sigma_1^2\,\mathrm{Tr}\left(\left\langle\boldsymbol{\Xi}_2^T\boldsymbol{\Xi}_2\right\rangle\right)\mathbf{x}_n^T\mathbf{x}_n\\ &= \sigma_1^2N_o\sigma_2^2\,\mathrm{Tr}(\mathbf{I})\,\mathrm{Tr}(\mathbf{x}_n\mathbf{x}_n^T)\\ &= \sigma_1^2N_o\sigma_2^2N_h\,\mathrm{Tr}(\mathbf{x}_n\mathbf{x}_n^T).\end{aligned} \tag{161}$$

Resubstitution into Equation (149) then yields

$$\begin{aligned}&\frac{1}{2P}\sum_{n=1}^{P}\left\langle\mathbf{b}^T\mathbf{b}\right\rangle = \frac{\sigma_{\mathbf{x}}^2}{2}\big(||\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1||_F^2 + \sigma_1^2N_i||\boldsymbol{\Omega}_2||_F^2 +\\ &\sigma_2^2N_o||\boldsymbol{\Omega}_1||_F^2 + \sigma_1^2N_i\sigma_2^2N_oN_h\big) +\\ &\frac{1}{2P}\big(\sigma_1^2||\boldsymbol{\Omega}_2||_F^2||\mathbf{X}||_F^2 + N_o\sigma_2^2||\boldsymbol{\Omega}_1\mathbf{X}||_F^2\\ &+ N_h\sigma_1^2N_o\sigma_2^2||\mathbf{X}||_F^2\big),\end{aligned} \tag{162}$$

Substituting Equations (146), (147) and (162) back into Theorem 5.6 concludes the proof. □

Proof of Theorem 5.2. Let $\boldsymbol{\xi}_n \in \mathcal{R}^{N_i}$ denote a random vector with independent and identically distributed, zero-centred entries with variance σ_x^2. Then the expected mean squared error over the training data when adding such a random perturbation to each input is

$$\begin{aligned}
&\left\langle \frac{1}{2P}\sum_{n=1}^{P} ||\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1(\mathbf{x}_n+\boldsymbol{\xi}_n)-\mathbf{y}_n||_2^2 \right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}\left\langle ||\underbrace{\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n-\mathbf{y}_n}_{a}+\underbrace{\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\boldsymbol{\xi}_n}_{b}||_2^2\right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}\mathbf{a}^T\mathbf{a}+\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\left\langle\mathbf{b}\right\rangle+\frac{1}{2P}\sum_{n=1}^{P}\left\langle\mathbf{b}^T\mathbf{b}\right\rangle .
\end{aligned} \tag{163}$$

The first summand is identical to Equation (146), the second summand is

$$\begin{aligned}
&\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\left\langle\mathbf{b}\right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}2(\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n-\mathbf{y}_n)^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\left\langle\xi_n\right\rangle \\
&= 0
\end{aligned} \tag{164}$$

and for the third summand, using Equations (53) and (57) we get

$$\begin{aligned}
\frac{1}{2P}\sum_{n=1}^{P}\left\langle\mathbf{b}^T\mathbf{b}\right\rangle &= \frac{1}{2P}\sum_{n=1}^{P}\left\langle\boldsymbol{\xi}_n^T\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\boldsymbol{\xi}_n\right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}\mathrm{Tr}\left(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\left\langle\boldsymbol{\xi}_n\boldsymbol{\xi}_n^T\right\rangle\right) \\
&= \frac{1}{2P}\sum_{n=1}^{P}\sigma_{\mathbf{x}}^2\,\mathrm{Tr}\left(\boldsymbol{\Omega}_1^T\boldsymbol{\Omega}_2^T\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\right) \\
&= \frac{\sigma_{\mathbf{x}}^2}{2}||\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1||_F^2 .
\end{aligned} \tag{165}$$

Substituting Equations (146), (164) and (165) back into Equation (163), then concludes the proof. □

Proof of Theorem 5.4. The expected mean squared error over the training data of a linear solution (Definition 2.3) under additive, independent and identically distributed zero-centred parameter noise with variance σ_1^2 and σ_2^2 is

$$\begin{aligned}
&\left\langle \frac{1}{2P}\sum_{n=1}^{P}||\left(\boldsymbol{\Omega}_2+\boldsymbol{\Xi}_2\right)\left(\boldsymbol{\Omega}_1+\boldsymbol{\Xi}_1\right)\mathbf{x}_n-\mathbf{y}_n||_2^2\right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}\left\langle||\underbrace{\boldsymbol{\Omega}_2\boldsymbol{\Omega}_1\mathbf{x}_n-\mathbf{y}_n}_{a}+\underbrace{\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\mathbf{x}_n+}_{b}\underbrace{\boldsymbol{\Xi}_2\boldsymbol{\Omega}_1\mathbf{x}_n+\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\mathbf{x}_n}_{b}||_2^2\right\rangle \\
&= \frac{1}{2P}\sum_{n=1}^{P}\mathbf{a}^T\mathbf{a}+\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\left\langle\mathbf{b}\right\rangle+\frac{1}{2P}\sum_{n=1}^{P}\left\langle\mathbf{b}^T\mathbf{b}\right\rangle .
\end{aligned}$$

The first summand is again identical to Equation (146) and the second summand is

$$\frac{1}{2P}\sum_{n=1}^{P}2\mathbf{a}^T\left\langle\mathbf{b}\right\rangle = 0 \tag{166}$$

since

$$\left\langle\mathbf{b}\right\rangle = \left\langle\boldsymbol{\Omega}_2\boldsymbol{\Xi}_1\mathbf{x}_n+\boldsymbol{\Xi}_2\boldsymbol{\Omega}_1\mathbf{x}_n+\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\mathbf{x}_n\right\rangle = \boldsymbol{\Omega}_2\left\langle\boldsymbol{\Xi}_1\right\rangle\mathbf{x}_n+\left\langle\boldsymbol{\Xi}_2\right\rangle\boldsymbol{\Omega}_1\mathbf{x}_n+\left\langle\boldsymbol{\Xi}_2\boldsymbol{\Xi}_1\right\rangle\mathbf{x}_n = 0 . \tag{167}$$

Which leaves us with the third term, which is identical to Equation (158) which equals

$$\frac{1}{2P}\sum_{n=1}^{P}\left\langle \mathbf{b}^T\mathbf{b}\right\rangle = \frac{1}{2P}\sum_{n=1}^{P}\mathbf{x}_n^T\left(\sigma_1^2||\mathbf{\Omega}_2||_F^2\mathbf{I} + N_o\sigma_2^2\mathbf{\Omega}_1^T\mathbf{\Omega}_1 + N_h\sigma_1^2N_o\sigma_2^2\mathbf{I}\right)\mathbf{x}_n$$
$$= \frac{1}{2}\left(\sigma_1^2||\mathbf{\Omega}_2||_F^2\operatorname{Tr}(\mathbf{\Sigma}_{xx}) + N_o\sigma_2^2\operatorname{Tr}(\mathbf{\Omega}_1^T\mathbf{\Omega}_1\mathbf{\Sigma}_{xx}) + N_h\sigma_1^2N_o\sigma_2^2\operatorname{Tr}(\mathbf{\Sigma}_{xx})\right)\,.$$

Substituting Equations (146) and (166) and **??** back into **??**, then concludes the proof. □