Neural Networks Trained to Solve Differential Equations Learn General Representations

Martin Magill, Faisal Z. Qureshi, & Hendrick W. de Haan
Faculty of Science, University of Ontario Institute of Technology, Oshawa
martinmagill.netlify.com
faculty.uoit.ca/dehaan/cNAB.LAB/

Overview

• In machine vision, CNN layers can be visualized as the features they learn to identify.

• Neural networks can learn the solutions to differential equations.

• **Question:** Do the layers in these networks encode useful information about the solution?

• **Answer:** Yes! For instance, the first layer identifies important regions of the input domain.

• **Bonus:** The same representations are learned reliably, even when the equations are modified.

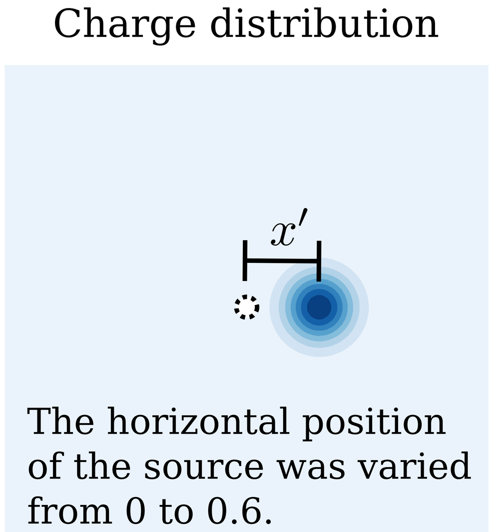
Family of problems

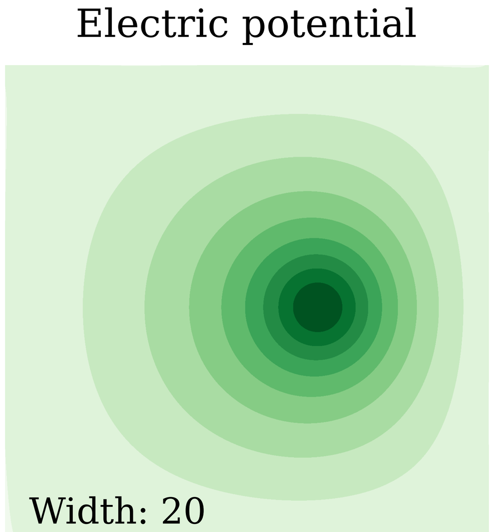
Used 4-layer fully-connected tanh neural networks to solve the **boundary value problem (BVP)**

$$\nabla^2 u(x,y) = s(x,y) \quad \text{for} \quad (x,y) \in \Omega,$$
$$u(x,y) = 0 \quad \text{for} \quad (x,y) \in \partial\Omega,$$
$$s(x,y) = -\frac{\exp\left(-\frac{(x-x')^2+(y-y')^2}{2r^2}\right)}{2\pi r^2},$$

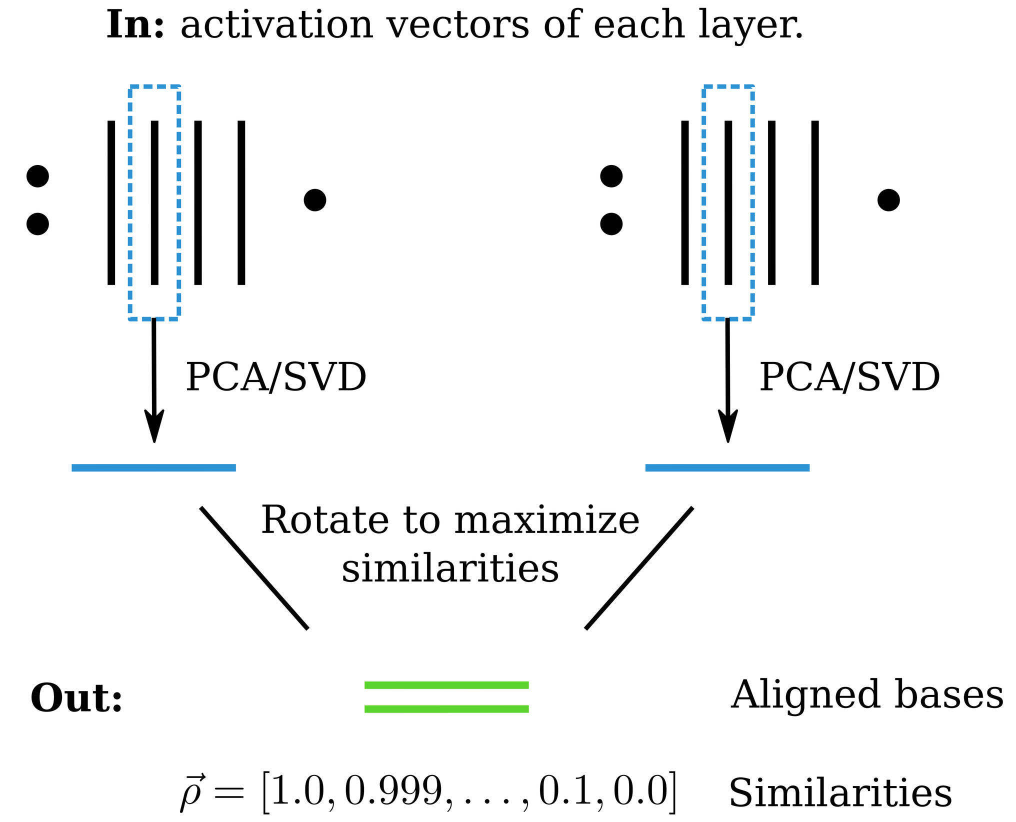
where Ω is a square domain.

This models the **electric potential** of a localized charge distribution on a square with grounded edges.

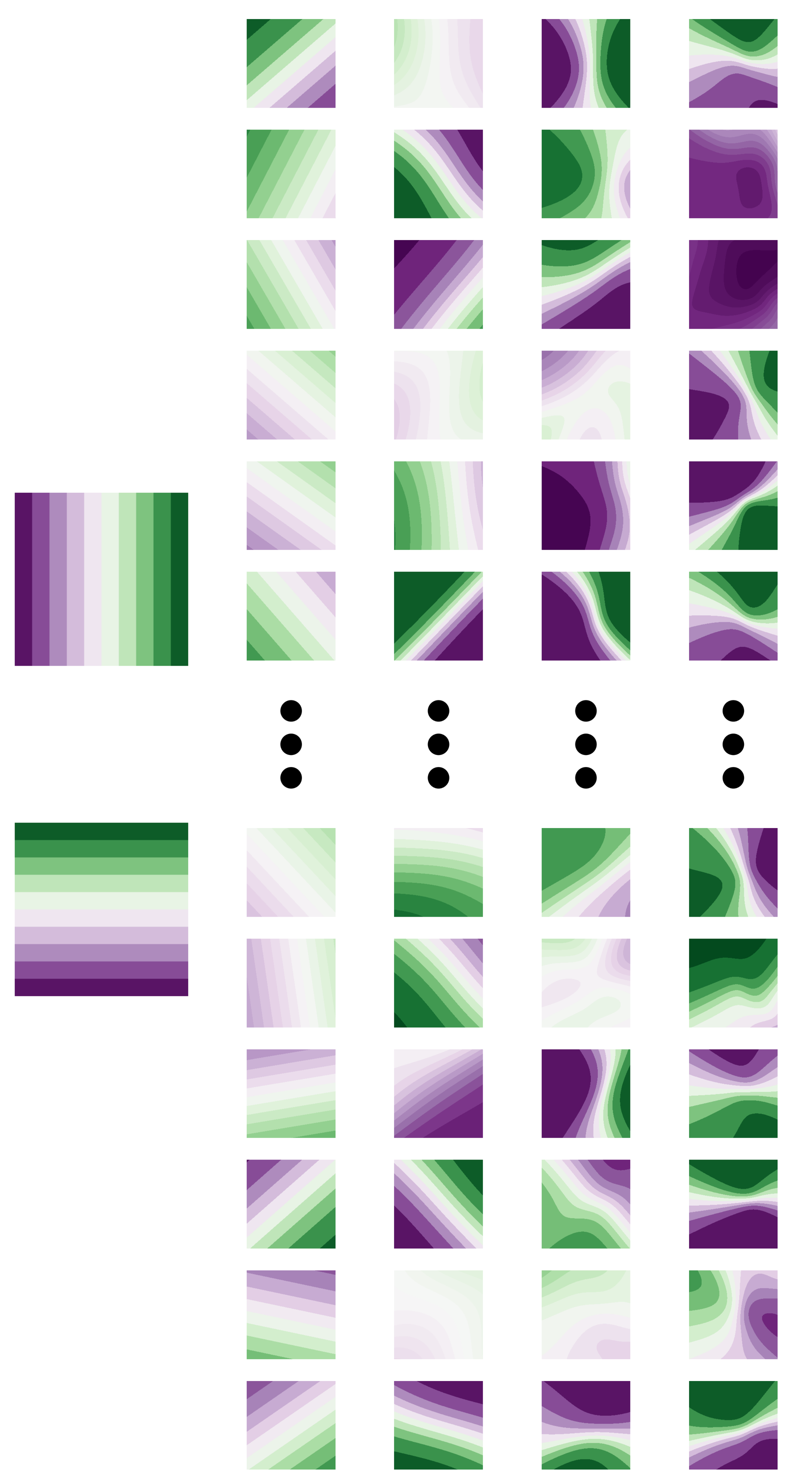




Layer-wise SVCCA



Interpreting the networks



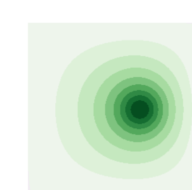
• **Inputs:** coordinates of a point (*x*,*y*).

• **Output:** estimated potential *u*(*x*,*y*).

• **Loss:** MSE of BVP equations.

• **Left:** Activation vectors of each neuron in a network trained at *x*'=0.3, shown as functions over the input domain.

• Note that it is difficult to interpret the activation vectors directly.



• **Right:** The same network after layer-wise SVCCA with a second network trained at *x*'=0.6.

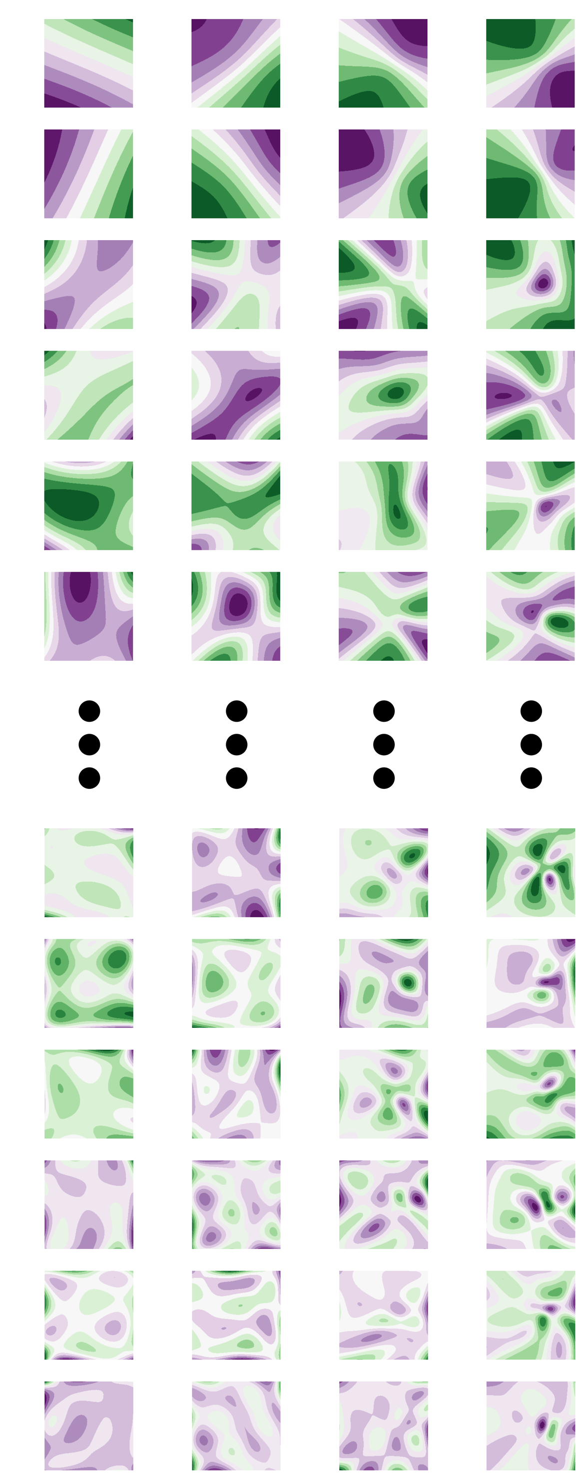
• Components are sorted from top to bottom by similarity scores.

• The components in the **first layer** accentuate the input regions that are important to both networks simultaneously.

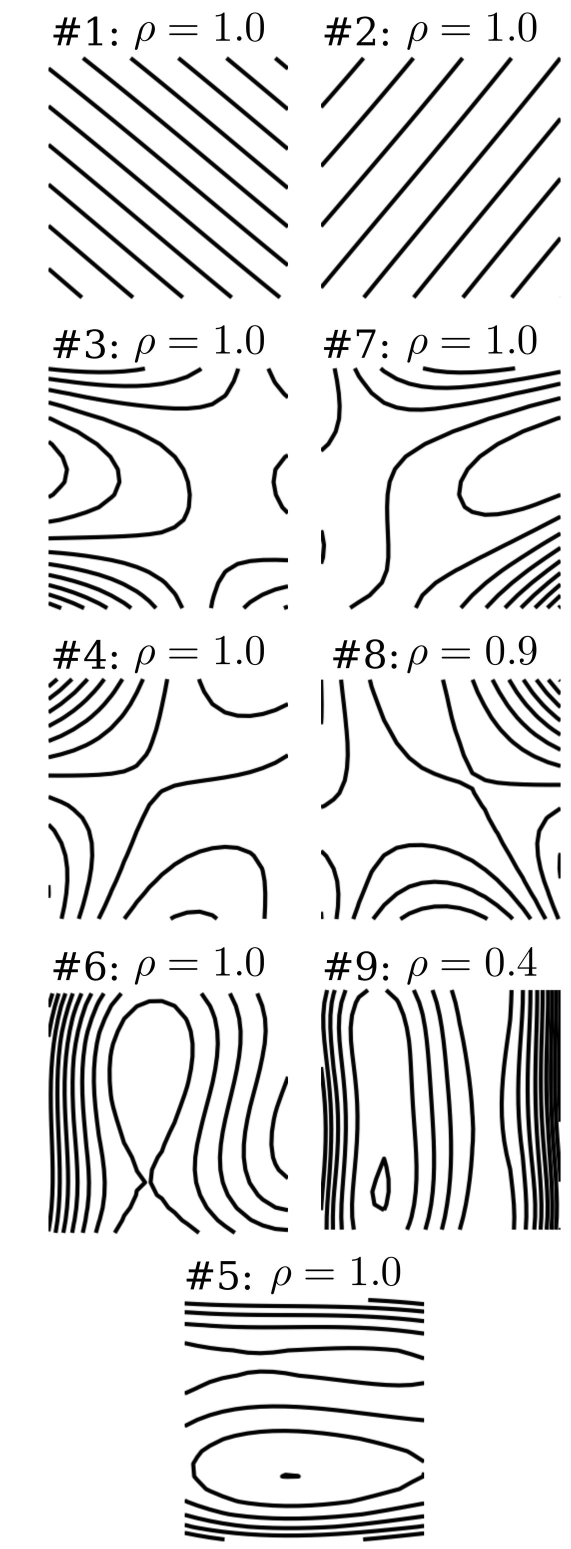
• The fourth component, for instance, highlights the top-left and bottom-right corners.

• The functions in the **last layer** form a basis that represents both outputs efficiently.

• In all layers, higher-order components become more multimodal, like Fourier modes.



The first layer learns coordinates



• **Left:** The nine leading components in the first layer of a network of width 192 trained at x'=0.6 after layer-wise SVCCA with itself.

• Labels show similarity values and their order when sorted by similarity.

• They **act as coordinates** over the input domain. The contour lines are densest where each coordinate is most sensitive.

• **First row:** These are simply rotations of the two **original coordinates**, x and y.

• **Second and third rows:** These four, together, show position relative to the **four corners** of the domain.

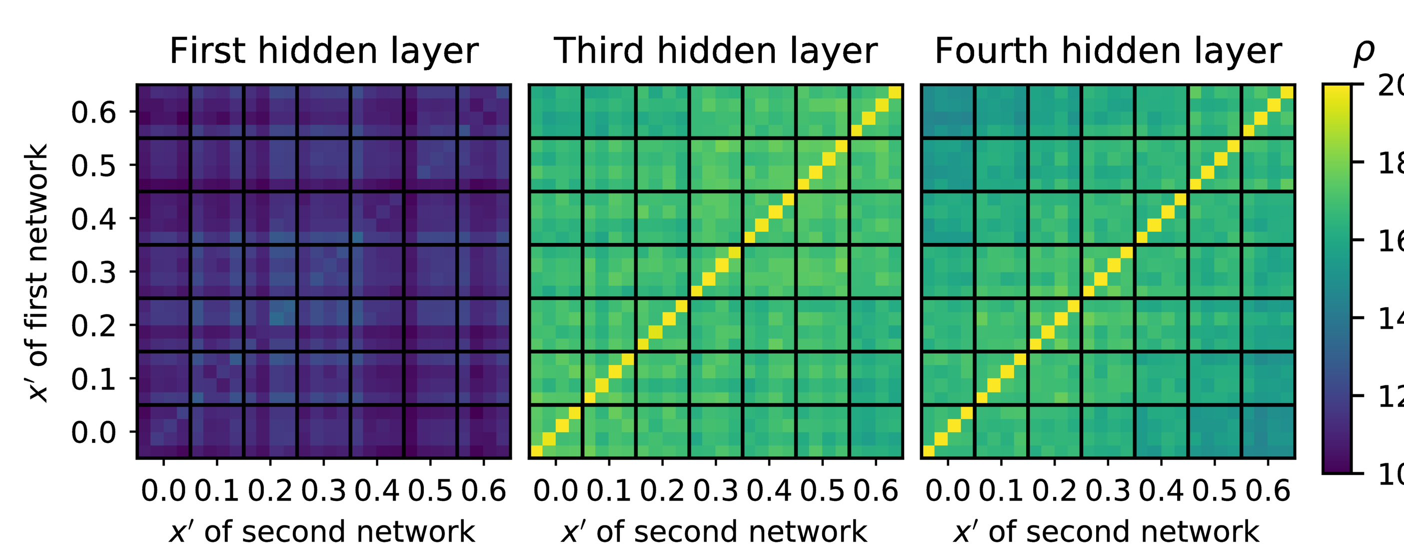
• **Fourth and fifth rows:** These capture distance to the **four walls** of the domain.

• **For all sufficiently wide networks,** the leading components of the first layer are mixtures of these features.

• This result is **reproducible** across different random initializations.

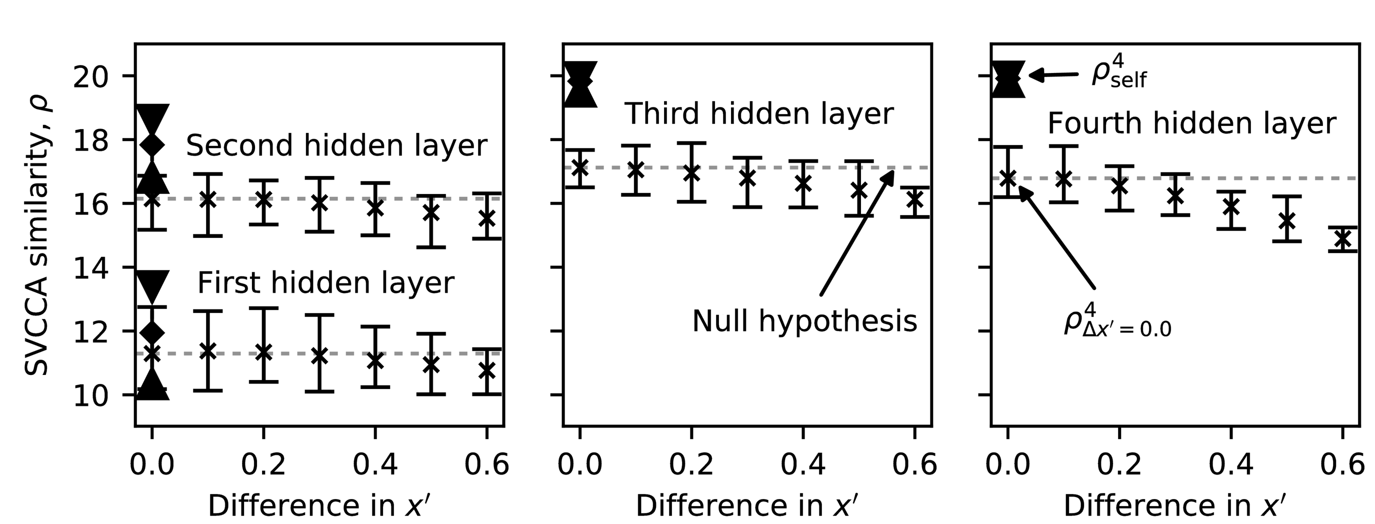
• It is also **general**, in that it does not depend on the *x*' of the two networks used for layer-wise SVCCA.

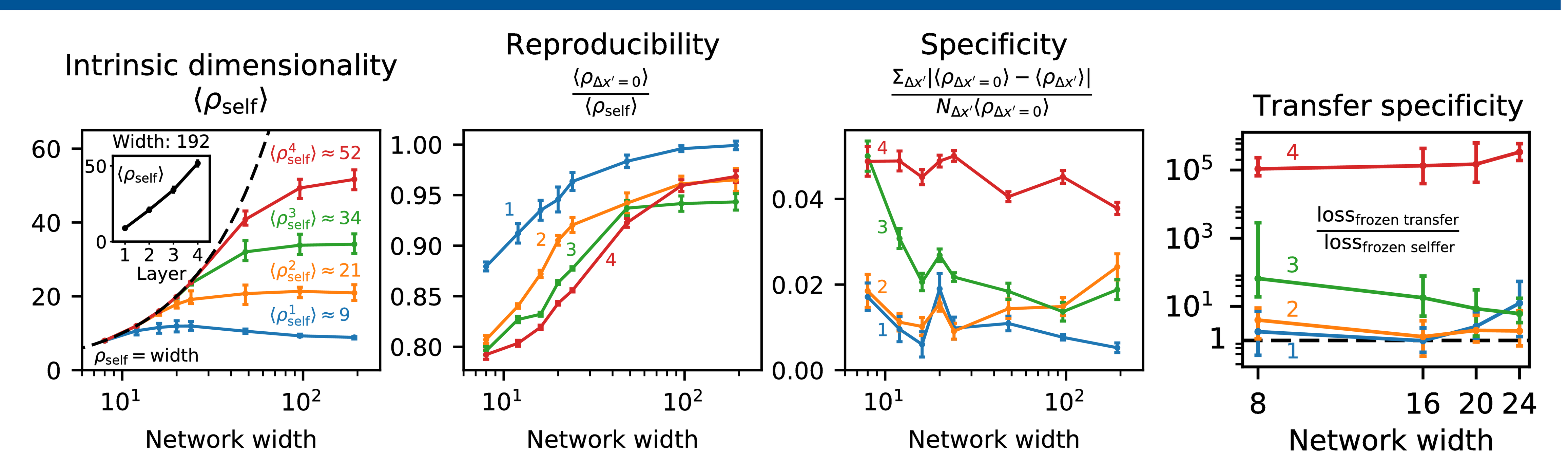
Quantifying layer specificity versus generality



• **Above:** Matrices of ρ, the sum of the SVCCA similarities, computed layer-wise between networks trained from different random seeds (between black lines) and at different *x*' values.

• **Below:** From the matrices, we extract the self-similarity ρ_{self}, the similarity $\rho_{\Delta x'=0}$ across random seeds at fixed *x*', and the similarity as a function of *x*', $\rho_{\Delta x'}$.





• The **intrinsic dimensionality** converges at high widths, as layers converge to finite-dimensional representations.

• Wide layers also have very high **reproducibility** across different random initializations.

• The **fourth layer** has **high specificity**, as its functional behaviour changes significantly when x' varies.

• The **first layer** has **low specificity**, because it learns a **general representation** that works well for all x'.

• The second layer is also quite general, but the third layer transitions from specific in narrow networks to general in wide networks.

• To validate our measure of specificity, we also measured specificity using an existing approach based on transfer learning tests (Yosinski et al. 2014, Adv Neural Inf Process Syst. 3320–3328).

• We found good agreement between the measures, and our method was orders of magnitude faster.