NEURAL COMPATIBILITY: FUNCTIONAL SIMILARITY WITH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

An ongoing problem for computer science and neuroscience is to chart the similarities and differences between how both artificial and biological neural systems represent data and process information. Representational similarity and representational alignment methods seek to quantify and close these gaps. A recent idea is to quantify whether representations from one model are usable by another model by "stitching" activity from one into the other. However, a fundamental property of neural systems is that they learn and adapt to new inputs. We therefore ask what makes two neural systems "compatible" in the sense that one model can efficiently *learn to use* representations from the other. We illustrate with simple examples how "correlation", "stitchability", and "compatibility" between models are in principle distinct ways of conceptualizing similarity between models and brains. Compatibility provides more evidence that representations are most similar across corresponding fractional depths. Despite the shown in-principle distinction, we find empirically that among pretrained image classification models, stitchability and compatibility are strongly correlated. We conclude that 'neural compatibility' is conceptually distinct from other notions of representational alignment, but identifying in-practice impactful differences requires further investigation.

1 INTRODUCTION

Converging towards a unified theory of information-processing in brains and neural networks promises not just more effective and efficient machine learning, but also a better scientific understanding of the brain. In recent years, tools for quantifying similarities and differences between neural representations have proliferated (for reviews, see Klabunde et al. (2025); Sucholutsky et al. (2024); Cloos et al. (2025)). Different methods make different assumptions and answer different questions. However, no current quantitative methods for comparing representations accounts for learning, despite the incredible capacity of neural systems to learn and adapt to new inputs, and despite the central role learning plays in shaping internal representations.

Consider the motivating problem of restoring sight or hearing to patients with irreparable damage to their eyes or ears. An artificial camera or microphone may provide surrogate input as *neural prosthetic*s by directly stimulating a blind or deaf patient's visual or auditory cortex (Abbasi & Rizzo, 2021; Wang et al., 2024; Fernández & Normann, 2017; Grant et al., 2016). Such surgical interventions have been attempted, and their efficacy depends on the brain's ability to learn and adapt to a new prosthetic source of stimulation (Shepherd et al., 2013; Abbasi & Rizzo, 2021; Fernández & Normann, 2017). According to Abbasi & Rizzo, neural prosthetic technology is hindered primarily by our limited understanding of neuroscience and brain plasticity rather than by device engineering. This raises the question: **what form of representation is most "compatible" with a (possibly biological) downstream model, taking into account that the downstream neural system will adapt and learn to its new inputs?**

Colloquially, the concept of two systems being compatible does not always involve adaptation or learning; a blood transfusion requires the donor's blood is compatible with the recipients (an asymmetric relationship), and a car engine is compatible with specific types of fuel. In contrast to these, when the recipient is a neural system, we argue that compatibility should be construed as the recipient effectively *learning to use* the donor's representations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

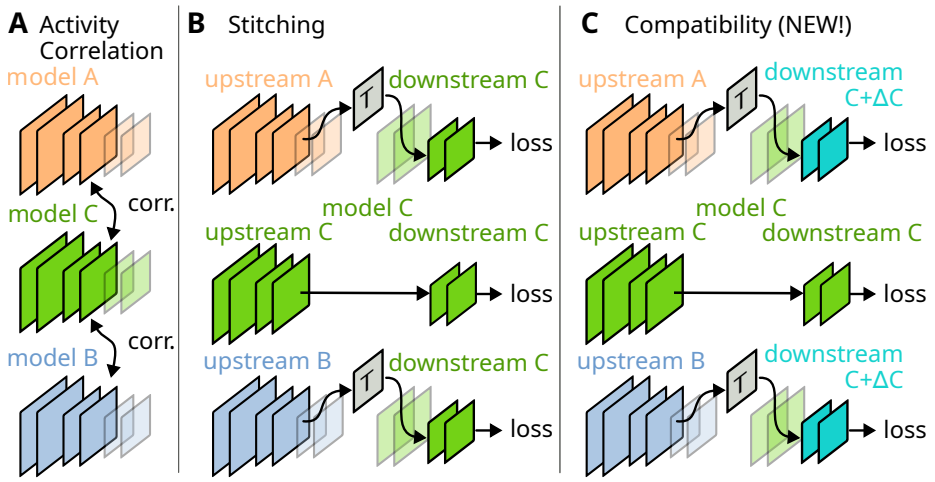

Figure 1: Three broad categories of methods for comparing neural representations (visualized as stacks of 2D feature maps). **A)** Activity correlation methods (RSA, CKA, CCA, etc.) correlate activity in hidden layers of each model, insensitive to downstream computations. Two layers are activity-correlated to the extent that they show analogous responses across many inputs. **B)** Stitching methods create a hybrid from initial upstream layers of one model, a simple mapping or "stitching layer" (gray T), and downstream layers of another model. Two layers are stitchable with respect to a desired behavior or task to the extent that this hybrid performs well. We compare stitchability of models A and B with respect to a reference downstream model C. **C)** We introduce "compatibility" as an extension of stitchability: a downstream model C is "compatible" with upstream A or B if it can effectively *learn to use* representations from the upstream model.

Learning is fundamental to neural systems. We show below that accounting for learning can (in principle) flip the answer to the question of which upstream system provides the best representations for a given downstream system. This makes neural compatibility (with learning) a qualitatively distinct kind of comparison between neural models, complementary to existing measures of representational or functional similarity. While this paper is largely theoretical, the practical implication is that designing optimal inputs for neural prosthetics — or any other hybrid neural system that is allowed to learn — is not merely a matter of reconstituting signals that are maximally "similar" to those that were present in the original system, nor does it reduce to existing notions of functional similarity such as stitchability.

We categorize existing work on neural similarity into two broad categories (these are non-exhaustive but capture the most relevant distinctions to our work):

- **Activity correlation** methods present identical stimuli to two neural systems and quantify statistical relationships between their internal activations (Figure 1A). Methods in this category include Representational Similarity Analysis (RSA) (Kriegeskorte, 2009; Diedrichsen & Kriegeskorte, 2017), Centered Kernel Alignment (CKA) (Kornblith et al., 2019), Canonical Correlation Analysis (CCA) (Hardoon et al., 2004; Raghu et al., 2017; Morcos et al., 2018), Normalized Bures Similarity (NBS) (Harvey et al., 2024a), and many variations on these. Detailed analysis of similarities and differences between methods within the activity correlation family is outside the scope of this work, but see (Williams et al., 2021; Shah et al., 2023; Harvey et al., 2024a; Klabunde et al., 2025; Sucholutsky et al., 2024; Cloos et al., 2025).

  Activity correlation methods quantify similarities and differences in neural representations purely in terms of those representations' relation to inputs (or their relation to each other across inputs (Edelman, 1998; Kriegeskorte & Diedrichsen, 2019)), regardless of what behavior the representation leads to downstream.

- **Stitching** methods, which are used by "functional similarity" tests, work by composing "upstream" layers from one model with "downstream" layers from another model and quantifying the performance of the resulting hybrid model (Figure 1B) (Lenc & Vedaldi, 2015; Bansal et al., 2021; Csiszárik et al., 2021; Hernandez et al., 2023). Unlike activity

correlation, stitching depends on how representations are *used* by the downstream system. However, prior work considers the downstream system to be fixed.

We propose **neural compatibility** as a third category of representational comparison which is qualitatively distinct from both activity correlation and stitching (Figure 1C). Our definition of compatibility builds on stitching: we say that two neural systems are "compatible" to the extent that they can be stitched together *and allowed time to learn and adapt to each other* and then achieve good behavior. While drawing inspiration from neuroscience, we illustrate the main ideas in this work by fine-tuning downstream neural networks. We prove a qualitative distinction between stitching and compatibility by constructing a set of three models (A, B, C) such that stitching A into C performs worse than stitching B into C, yet A and C are more compatible than B and C.

The remainder of this paper is structured as follows: Section 2 uses simple low-dimensional representations and hand-crafted networks to prove qualitative distinctions between activity correlation, neural stitching, and neural compatibility. In Section 3, we conduct an empirical study of compatibility of pretrained convolutional networks on benchmark vision tasks. Section 4 concludes.

2 Distinguishing activity-correlation, stitchability, and compatibility

All of these methods consider a pair of models and a layer or group of neurons from each. Let A and C be two neural network models, and let ℓ_A and ℓ_C identify layers or regions from them. Each model is a function mapping stimuli $\boldsymbol{s} \in \mathcal{S}$ to outputs $\mathbf{y} \in \mathcal{Y}$ via a sequence of intermediate representations $\mathbf{x}^\ell \in \mathcal{X}^\ell$. We will assume deep non-recurrent models for simplicity so that we can separate each system into an upstream sub-model $A^{\ell_A}_{\wedge} : \mathcal{S} \to \mathcal{X}^{\ell_A}$ and a downstream sub-model $A^{\ell_A}_{\vee} : \mathcal{X}^{\ell_A} \to \mathcal{Y}$ such that the original model is expressible as function composition $A = A^{\ell_A}_{\vee} \circ A^{\ell_A}_{\wedge}$.

We say that ℓ_A and ℓ_C are **activity-correlated** to the extent that there exists a simple mapping from activity in one to activity in the other across the distribution of inputs (Klabunde et al., 2025; Sucholutsky et al., 2024). To make the dependence on the input distribution explicit, let p denote a probability distribution on $\mathcal{X}$. Formally, activity correlation methods depend on $(p, A^{\ell_A}_{\wedge}, C^{\ell_C}_{\wedge})$ and possibly some other method-specific parameters. For example, CKA computes

$$\text{correlation}(p, A^{\ell_A}_{\wedge}, C^{\ell_C}_{\wedge}) = \frac{\|\boldsymbol{\Sigma}_{AC}\|_{\text{F}}^2}{\|\boldsymbol{\Sigma}_{AA}\|_{\text{F}}\|\boldsymbol{\Sigma}_{CC}\|_{\text{F}}}\,, \tag{1}$$

where $\boldsymbol{\Sigma}_{AC}$ is the cross-covariance between features from ℓ_A and features from ℓ_C and $\|\cdot\|_{\text{F}}$ is the Frobenius norm, all of which may optionally be computed in the feature space of a nonlinear kernel, though it is common practice to directly use the cross-covariance of the neural activity (Cortes et al., 2012; Kornblith et al., 2019). Importantly, these covariance and cross-covariance terms depend only on terms computed by sampling inputs $\mathbf{s} \sim p$ and running them through upstream parts of models $A^{\ell_A}_{\wedge}$. Other methods like CCA, RSA, or NBS could be used instead of CKA without affecting our qualitative conclusions, so we will use the more general term "correlation."

We say that ℓ_A is **stitchable** into ℓ_C to the extent that a hybrid model,

$$C^{\ell_C}_{\vee} \circ T \circ A^{\ell_A}_{\wedge}\,,$$

composing the upstream part of A, a "stitching layer" T selected from a family of transforms $T \in \mathcal{T}$, and the downstream part of C, performs well on a relevant benchmark, optimized only with respect to the parameters of T (Lenc & Vedaldi, 2015; Bansal et al., 2021; Csiszárik et al., 2021). Formally, stitchability depends on $(p, A^{\ell_A}_{\wedge}, \mathcal{T}, C^{\ell_C}_{\vee}, \text{loss})$ where now p is a joint distribution on inputs and targets to compute some supervised loss. We then define stitchability by the best-case loss

$$\text{stitchability}(p, A^{\ell_A}_{\wedge}, \mathcal{T}, C^{\ell_C}_{\vee}, \text{loss}) = -\min_{T \in \mathcal{T}} \mathbb{E}_{\mathbf{s},\mathbf{y}\sim p}\left[\text{loss}(C^{\ell_C}_{\vee}(T(A^{\ell_A}_{\wedge}(\mathbf{s}))), \mathbf{y})\right]\,. \tag{2}$$

Loss is negated so that larger numbers correspond to "greater stitchability."

Extending stitchability, we say that ℓ_C is **compatible** with ℓ_A if the hybrid model can feasibly learn good performance on the task after stitching. Essentially, compatibility equals stitching plus fine-tuning. We are especially interested in the case where $A^{\ell_A}_{\wedge}$ is static and $C^{\ell_C}_{\vee}$ must *learn to*

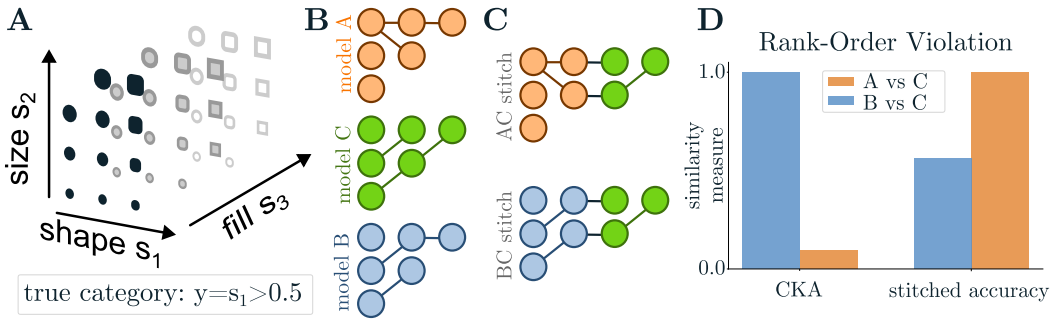

Figure 2: Minimal example illustrating the qualitative distinction between activity correlation and stitchability. **A)** A 3-feature space where all are positively correlated with each other and the task is $y = s_1 > 0.5$ ("squares" vs "circles"). **B)** Three one-hidden-layer linear networks; circles are units and lines are weights. Input nodes are $[s_1, s_2, s_3]$. Model A uses the true category-defining feature s_1. Models B and C each use the other features. **C)** Stitched models. **D)** By any activity-correlation method, the hidden layers of B and C are identical, while A is merely correlated with C. By stitching, the AC model achieves perfect performance while the BC model's outputs only weakly correlate with labels. Quantitative details can be found in Appendix A.1.

use representations coming from A, since this is analogous to the brain learning to use a prosthetic sensory device, but one could also consider end-to-end compatibility or compatibility just adapting the upstream model (we leave this for future work). Formally, compatibility depends on $(p, A^{\ell_A}_{\wedge}, \mathcal{T}, C^{\ell_C}_{\vee}, \text{loss}, \Lambda, \beta)$, where Λ is the learning algorithm used to fine-tune the hybrid model after stitching, and β is a non-negative "budget" constraining the amount of fine-tuning. We will use gradient descent for a fixed number of steps, but in general compatibility should consider how the component models learn and what the associated costs are. Our qualitative arguments should hold regardless of how the systems learn. Compressing notation, let $H_\theta : \mathcal{S} \to \mathcal{Y}$ be the stitched hybrid model with trainable parameters θ; in our experiments, θ contains the parameters of $C^{\ell_C}_{\vee}$ and the parameters of the stitching layer T. We initialize θ_0 from the best stitched model according to equation 2, then we train for a budget of β steps to arrive at θ_β. This gives the following formal definition of compatibility when the learning algorithm Λ is SGD:

$$\text{compatibility}(p, A^{\ell_A}_{\wedge}, \mathcal{T}, C^{\ell_C}_{\vee}, \text{loss}, \text{SGD}, \beta) = -\mathbb{E}_{\mathbf{s},\mathbf{y}\sim p}\left[\text{loss}(H_{\theta_\beta}(\mathbf{s}), \mathbf{y})\right] \tag{3}$$

Note that by our definition, compatibility equals stitchability whenever $\beta = 0$, since we treat the initial optimization of the stitching layer in equation 2 as free, only counting learning costs to further adapt the whole system. A better analogue of the neural prosthetic case which could be explored in future work may be to take T to be initialized in other ways, perhaps randomly or to maximally correlate with $C^{\ell_C}_{\wedge}$. In this work, matching stitchability and compatibility when $\beta = 0$ makes for a sharper comparison between the two.

Equations (1), (2), and (3) make commitments of the three approaches explicit, but they are verbose. Below, we will drop terms and write "compatibility$(A_\wedge, C_\vee)$" when the data distribution, layers, task, stitching family, and learning budget are implied by the context.

2.1 A MINIMAL EXAMPLE DISTINGUISHING ACTIVITY CORRELATION FROM STITCHABILITY

We are interested in what drives *qualitative* differences between activity correlation, stitchability, and compatibility. Therefore we begin with simple examples that demonstrate these differences by wiring up three models, A, B and C to show case *rank order violations* where, for instance,

$$\begin{aligned} &\text{correlation}(A_\wedge, C_\wedge) > \text{correlation}(B_\wedge, C_\wedge)\,, \text{ but} \\ &\text{stitchability}(A_\wedge, C_\vee) < \text{stitchability}(B_\wedge, C_\vee)\,. \end{aligned} \tag{4}$$

Such a rank-order violation demonstrates that the underlying quantitative comparisons between models have some deeper *qualitative* distinction in what they are sensitive to.

This section demonstrates a qualitative distinction between activity-correlation methods and stitchability. Prior work has argued for this distinction and shown it empirically in larger networks (Bansal

et al., 2021; Csiszárik et al., 2021; Hernandez et al., 2023; Braun et al., 2025); our goal here is to distill the core ideas using a minimal example that can be worked out on paper and to set up the distinction between stitchability and compatibility in the following section.

Figure 2 illustrates a simple input feature space $\mathbf{s} \in [0,1]^3$ where the axes of the space correspond to three properties of a set of objects; s_1 is shape, s_2 is size, and s_3 is shading of input objects, in arbitrary units (Figure 2A. The task is categorize y = shape > 0.5 invariant to the other features. To keep the models as simple as possible (but no simpler), A, B, and C are two-layer linear networks with 3 input units, 2 hidden units, and one output (Figure 2B). By construction, B and C have identical upstream layers and so they are maximally activity-correlated by any measure of activity correlation. However, both B and C neglect the critical task-relevant feature s_1, but this feature is encoded in the hidden layer of A, and so stitching A into C achieves perfect performance where stitching B into C does not (Figure 2D). Thus, this example has illustrated rank-order violation — as in equation 4 — which demonstrates a qualitative distinction between *all* activity correlation methods and stitchability.

This argument is in line with recent work contrasting representational similarity with functional similarity (Bansal et al., 2021; Braun et al., 2025; Theiss et al., 2026; Smith et al., 2025). Our example distills the principle down to minimal models. However, we note that "functional similarity" is broader than stitchability, and different works define functional similarity in different ways.

2.2 What drives compatibility above and beyond stitchability?

We define neural compatibility as an extension of stitchability with additional fine-tuning after stitching is complete. Clearly, then, models must be at least as compatible as they are stitchable. But what drives compatibility above and beyond stitchability? In the following Section 2.3, we will illustrate a minimal but nontrivial example of rank-order violation between stitchability and compatibility. Let us first discuss what makes an example of compatibility trivial or interesting. One might, for instance, hope for another equally simple linear example as in the previous section, but the following Lemma essentially rules this out:

Lemma 1. *Assume that (i) the downstream system $C_\vee$ is affine, i.e. $C_\vee(\mathbf{x}) = \boldsymbol{W}_C\mathbf{x} + \boldsymbol{b}_C$; and (ii) the stitching family $\mathcal{T}$ is likewise affine, i.e. $T(\mathbf{x}) = \boldsymbol{W}_T\mathbf{x} + \boldsymbol{b}_C$. Define $(\boldsymbol{W}_{opt}, \boldsymbol{b}_{opt})$ to be the best affine readout of some upstream model's $\mathbf{x}^{\ell_A}$ for the task loss of interest, minimizing $\mathbb{E}[loss(\boldsymbol{W}\mathbf{x}^{\ell_A} + \boldsymbol{b}, \mathbf{y})$. Then, stitchability and compatibility are equivalent as long as the column span of $\boldsymbol{W}_{opt}$ is contained in the column span of $\boldsymbol{W}_C$.*

The proof follows simply from the fundamental theorem of linear algebra (Strang, 2016), which states that $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$ has a solution $\boldsymbol{w}$ if and only if $\boldsymbol{b}$ is in the column span of $\boldsymbol{A}$. Intuitively, if $C_\vee$ is constrained to be linear, then anything that can be gained by finetuning $C_\vee$ on new input features could have instead been gained by optimizing the (linear) stitching layer itself. Since we take optimization of the stitching layer itself to be zero-cost in equation 3, this leaves nothing more to learn by fine-tuning. For scalar outputs, any nontrivial ($\boldsymbol{W}_C \neq \mathbf{0}$) downstream readout spans the space, but for higher-dimensional output spaces, it is worth noting that downstream learning may still have an effect if $\boldsymbol{W}_C$ does not already span the best subspace of $\mathcal{Y}$.

Lemma 1 thus does point to a first "trivial" way in which compatibility and stitchability give different quantitative values even for linear models: stitching is limited to the functional basis provided by $C_\vee$, but further fine-tuning can rotate this basis to span new parts of the output domain $\mathcal{Y}$. However, this effect cannot be responsible for any rank-order violation between stitchability and compatibility, since restrictions on the output span of a model A/C hybrid would also apply to a model B/C hybrid.

Lemma 1 further implies that more interesting examples will be found only by taking $C_\vee$ to be a multi-layer network with the capacity to express nonlinear functions of the hidden layer activity. Taken to the extreme, what if $C_\vee$ is a very high-capacity model family? This can lead to rank-order violations between stitchability and compatibility in a trivial sense. If the learning budget β is large, then a high-capacity downstream model can re-learn to use arbitrary inputs. In such cases, arbitrating among upstream models largely becomes a matter of selecting the upstream model which provides the maximum amount of information about the stimulus. If $\mathrm{MI}(\mathbf{s}, A_\wedge(\mathbf{s})) >> \mathrm{MI}(\mathbf{s}, B_\wedge(\mathbf{s}))$ (where MI is Mutual Information), then the downstream model may learn to use

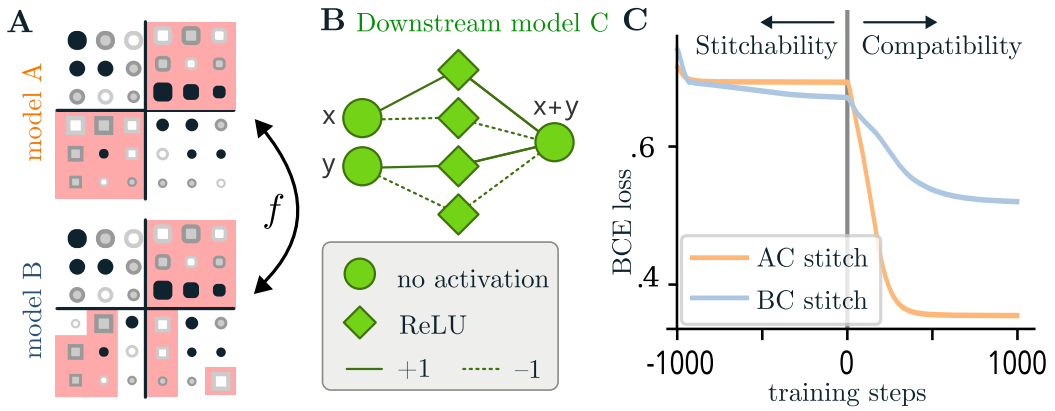

Figure 3: Minimal example illustrating the qualitative distinction between activity correlation and stitchability. **A)** Hidden layer activity of two candidate upstream models A and B, each with a 2D feature space embedding of stimuli from Figure 2. Red and white background indicates embeddings of stimuli where y $= 1$ and y $= 0$, respectively. They are related to each other by an invertible function $\mathbf{x}_B = f(\mathbf{x}_A)$. **B)** Architecture and initial weights of the downstream model $C_\vee$. Note that $C_\vee([x, y]) = x + y$, so the downstream model functionally implements a linear readout of its inputs despite the presence of ReLU nonlinearities in its middle layer. **C)** Binary Cross Entropy (BCE) loss curves when stitching representations from A or B into a frozen C (first 1000 steps) and then adapting C to its new inputs (last 1000 steps). The crossover of the loss curves during the latter "compatibility" phase demonstrates a rank-order violation between stitchability and compatibility.

representations from $A_\wedge$ more effectively simply because they contain more information, even if that information was inaccessible through stitching.

The fact that information content matters is both practically important but theoretically uninteresting. All else being equal, it implies that neural prosthetic systems that provide high-capacity or high-dimensional information about the world are preferred to low-capacity or low-dimensional inputs. However, not all else is necessarily equal. The argument so far misses an important point about the *format* of information: what if model A's representations contain maximal information about the input, but *extracting* that information is too burdensome and the downstream model C cannot feasibly learn to use it (Xu et al., 2020; Finzi et al., 2026)? Information being in a format such that a better downstream C is *feasibly learnable* is essential, and motivates our study of compatibility in the following section in terms of a finite learning budget while comparing upstream models that differ only in information format rather than amount of information.

2.3 A MINIMAL EXAMPLE DISTINGUISHING STITCHABILITY FROM COMPATIBILITY

The above considerations motivate our minimal example in Figure 3. We do not directly specify the stimulus $\mathbf{s}$, instead we construct the outputs of $A_\wedge^{\ell_A}$ and $B_\wedge^{\ell_B}$ (i.e. the hidden-layer representations in A and B which are then stitched into C), visualized in Figure 3A. In this example, we enforce that $A_\wedge$ and $B_\wedge$ have matched information about inputs by enforcing that there is an invertible function mapping between them. This example thus emphasizes that the *format* of information can also drive compatibility above and beyond differences in the *magnitude* of information.

Figure 3A imagines two upstream models $A_\wedge$ and $B_\wedge$, each providing two inputs to a downstream $C_\vee$. The downstream model has the capacity to learn nonlinear things, including perfectly solving the XOR problem present in the representations provided by $A_\wedge$, but at initialization its weights are such that $C_\vee$ is functionally equivalent to a linear model (Figure 3B). The best-possible linear stitching of $B_\wedge$ into $C_\vee$ achieves $> 50\%$ accuracy on the task, exploiting the fact that the x coordinate of model B's activation space correlates with the label. Meanwhile, no linear stitch of $A_\wedge$ into $C_\vee$ ever surpasses 50% chance performance. However, when $C_\vee$ is unfrozen and allowed to adapt to its new inputs, it *learns to use* representations from $A_\wedge$ much more effectively than those from $B_\wedge$ (Figure 3C). In other words, this example illustrates a simple case where

$$\text{stitchability}(A_\wedge, C_\vee) < \text{stitchability}(B_\wedge, C_\vee)\,, \text{ but}$$
$$\text{compatibility}(A_\wedge, C_\vee) > \text{compatibility}(B_\wedge, C_\vee)\,,$$

which is a rank-order violation between any stitchability and compatibility after just a few steps of fine-tuning. We emphasize that this example relies entirely on the difference in *format* between $A_\wedge$ and $B_\wedge$ rather than a difference in information content. Appendix A.3 formalizes this argument.

More generally, this example illustrates that compatibility depends on the interaction between representational format, the initial state of each part of system, the learning algorithm used by the systems, and associated costs and budgets. Learnability depends not just on the model architectures and parameters, but also on the learning algorithm. Which representations are most compatible with a downstream gradient-descent-based learning system may be different from what is most compatible with a downstream biological system. Identifying what is "learnable" by biological neural networks is a critical open problem for the design of neural prosthetics (Abbasi & Rizzo, 2021).

3 Experiment: Compatibility Across ImageNet ResNet Architectures

Given the *in principle* distinction between neural stitchability and neural compatibility demonstrated in the previous section, we wondered whether such a distinction would play out in pretrained neural networks. To explore this, we evaluated and compared stitchability and compatibility between three ResNet models (He et al., 2016) from the torchvision library[1], pretrained on ImageNet (Deng et al., 2009). Following previous work (Bansal et al., 2021; Csiszárik et al., 2021), we split each model at block-level outputs, i.e. at the skip connections. We analyzed hybrid models comprising all pairs of upstream and downstream blocks, not just those that have comparable relative depths (Hernandez et al., 2023).

For each pair of upstream and downstream model components, we first evaluated stitchability using a 1x1 Convolution as the stitching family (Bansal et al., 2021; Csiszárik et al., 2021). In the case that the shape between the upstream output and downstream input differ in height or width, we apply a fixed resize operation across the spatial dimensions of the upstream tensor to match the expected downstream shape (Hernandez et al., 2023). As all models were pretrained, we disabled further BatchNorm adaptation by merging BatchNorm parameters into convolution parameters.

It is critical in this phase of the experiment that the loss minimization with respect to the stitching layer T is carried out to convergence; poor optimization of T make models appear "unstitchable," and further optimization of T during fine-tuning may then make them appear "compatible," but this would merely reflect poor execution of the initial stitching rather than a difference between stitching and compatibility. We used two methods to optimize learning of the stitching layer T. First, we initialized the parameters of the 1x1 convolution using least-squares regression to best match the original activations from $C_\wedge$ (Csiszárik et al., 2021). Next, we searched for an initial learning rate $\lambda_0 \in [1e-9, 1e-1]$ giving the fastest rate of improvement for each hybrid model (Smith, 2017). We found that the ideal learning rate varied considerably depending on the upstream and downstream components involved. Throughout, all optimization was carried out using the Adam optimizer (Kingma & Ba, 2017) to minimize cross-entropy loss on ImageNet (Deng et al., 2009). The learning rate was decayed on a schedule $\lambda_t = 100\lambda_0/(100+t)$ where t is the optimization step.

After stitching converged, we next assessed compatibility by allowing the downstream model to further adapt to its new inputs. We further fine-tuned both the stitching layer T and the downstream component of the hybrid model $C_\vee$ with a budget of 1000 optimization steps. To ensure fair comparisons, we ran a new search for the optimal λ_0 at the start of the fine-tuning phase.

To our knowledge, only one previous study systematically analyzed stitchability between *all* pairs of blocks at all depths (Hernandez et al., 2023), though their results were limited to CIFAR10 and primarily shallower models. In agreement with their results, we find that "balanced" hybrid models — those where both upstream and downstream models are split at the same fractional depth — achieve the best stitching performance (green ellipse in Figure 4A,B, left). This also agrees with existing results using activity-correlation methods: layers at the same relative depth tend to be most similar across models (Kornblith et al., 2019; Nguyen et al., 2021; Lange et al., 2023). See Figure 8 for results for all pairs of blocks across all models. However, our results did not always reproduce the high stitching performance in the lower-triangle (late upstream blocks stitched into early

[1] `https://docs.pytorch.org/vision/main/models.html`

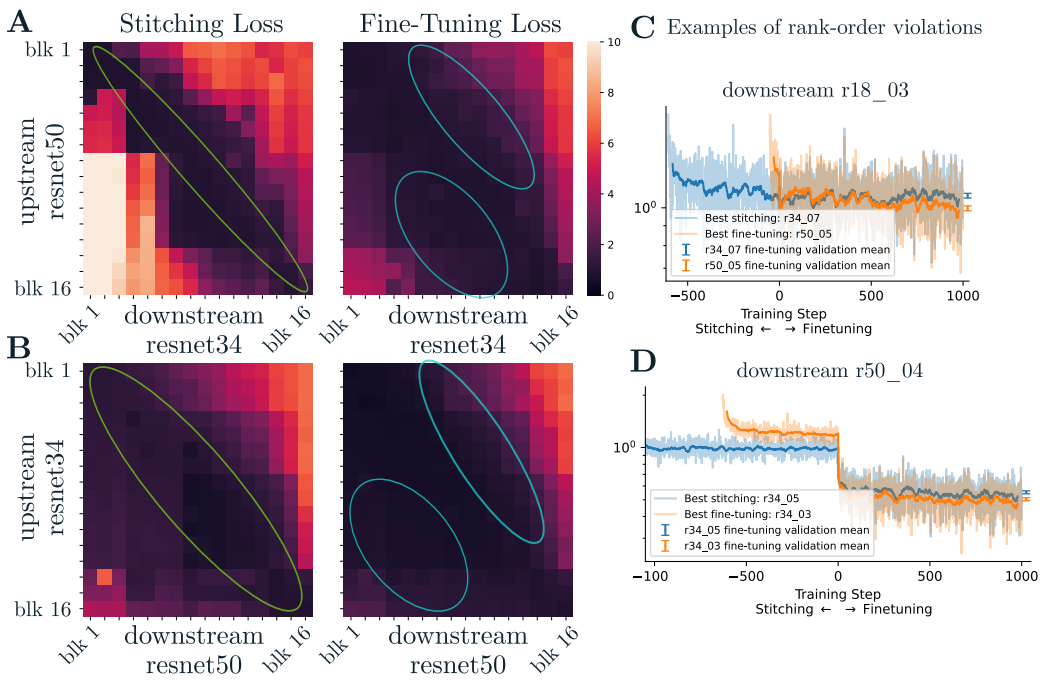

Figure 4: Results comparing stitching and compatibility across ResNet architectures. Layer names are abbreviated "r{ResNet-depth}_{block}". **A)** Heatmap of cross-entropy loss after stitching (left) and after fine-tuning (right) for all pairs of blocks where resnet50 supplies upstream and resnet34 supplies downstream layers. The green ellipse marks "balanced" models where the upstream and downstream models are divided at nearby fractional depths. Cyan ellipses call attention to non-balanced models that perform nearly as well as balanced models after fine-tuning. **B)** Same as (A) but for upstream resnet34 and downstream resnet50. **C)** A potential rank-order violation among a triple of models (compare to Figure 3C). The loss after stitching (step < 0) is lower for the blue model than the orange model (on the validation set, not pictured), while loss after fine-tuning (step ≥ 0) is lower for the orange than the blue. Shaded curves show training loss each step and bold lines show the running average. Errorbars after step 1000 indicate performance on the validation set after fine-tuning. **D)** Same as (C) for a second pair of models that did not exhibit a statistically-significant rank-order violation.

downstream blocks) found by Hernandez et al. (2023), suggesting that this phenomenon may be less general than originally thought.

Downstream finetuning improved the performance of all hybrid models, as expected. Interestingly, fine-tuning did not significantly improve performance for balanced hybrid models, suggesting that stitching already achieves near-optimal performance in these cases. For hybrid models located around the balanced diagonal (cyan ellipses in Figure 4A,B), we found that further fine-tuning resulted in performance comparable to the balanced stitched models. This shows that "compatible" models are indeed a superset of "stitchable" models. However, this alone does not yet demonstrate any *qualitative* distinction or rank-order violation between stitchability and compatibility.

We next evaluated model triples (upstream A, upstream B, downstream C) to see if any such rank-order violations occurred. For each downstream architecture (all three resnets split at all blocks), we identified the best upstream model after stitching and after fine-tuning according to validation loss. We checked for cases where the best fine-tuning hybrids and the best stitching hybrids came from different upstream models. Across 40 downstream models, we identified 5 significant cases of violations between stitchability and compatibility (Table A.5.1). In all cases, the effect was modest; the largest effect we see is an increase from $75.91 \pm 0.38\%$ to $78.41 \pm 0.36\%$ top-1 accuracy on the ImageNet validation set.

4 DISCUSSION

We argued for a three-way distinction between families of methods for comparing neural representations: activity-correlation, stitchability, and compatibility. The first two are sometimes called representational and functional similarity, respectively. The third — neural compatibility — is new. Compatibility is inspired by considering the problem of designing neural prosthetics (Shepherd et al., 2013; Abbasi & Rizzo, 2021; Fernández & Normann, 2017), as well as by recent work exploring "learnability" in defining features of neural representations (Morrison, 2024; Baker et al., 2026).

Considering rank-order-violations among triples of models (upstream A, upstream B, downstream C), we developed a minimal example that proves a *qualitative* distinction between activity-correlation and stitching (Bansal et al., 2021; Csiszárik et al., 2021; Braun et al., 2025), as well as a second minimal example proving a qualitative distinction between stitchability and compatibility. We emphasize that the latter example isolated the role of how representational *format*, rather than mere information content, can affect conclusions about which upstream system is most compatible with a given downstream system.

While our example in Section 2.3 showed that *in principle* there can be dramatic differences between stitchability and compatibility, our experimental results on pretrained ResNets showed only weak effects. The gap between stitchability and compatibility is likely much larger than this in practice for our motivating case of neural prosthetics. The key insight remains: when choosing the "best" upstream model to inject neural activity into a downstream model (which could be a biological brain), failing to account for *how the downstream system will adapt to its new inputs* could result in a suboptimal choice of upstream architecture. The ideal neural representations from the donor must simultaneously carry as much information about the stimulus as possible while also being *formatted* in such a way that the downstream system can learn to use it.

4.1 LIMITATIONS AND FUTURE WORK

The primary limitation of our work is that we restricted our experimental analysis to pretrained ResNets that were already highly similar before stitching them together into hybrids. We expect that much stronger empirical demonstrations of the difference between stitchability and compatibility will come from wiring together models with disparate training histories, modalities, or architectures. Techniques for stitching across architectures such as stitching convnets into vision transformers (or vice versa) are still under development (cf. Traft (2025); Peng et al. (2021)) but will be an interesting area for future work.

A second limitation, from the perspective of neural prosthetics, is that we consider stitching optimization itself to be "free." With current technology, neural prosthetic hardware cannot receive learning signals directly from the brain to fine-tune the upstream processing or the "stitching" per se. The brain must adapt to something more like a random stitch or a regression-optimized stitch rather than a task-optimized stitch. We opted to compare compatibility with task-optimized stitching to make a sharper argument for the importance of downstream learning; we expect that weaker stitching methods will only further highlight the qualitative difference between stitchability and compatibility.

Finally, we note that many existing representational comparison methods treat "usability" of the representation as key to defining what is represented (Boix-Adsera et al., 2022; Harvey et al., 2024b; Kriegeskorte & Diedrichsen, 2019; Almudévar & Ortega, 2026). Our work can be seen as taking "downstream learnability" as a particular kind of usability of representations (Baker et al., 2026). As formal frameworks for studying "usable information" advance (Xu et al., 2020; Almudévar & Ortega, 2026; Finzi et al., 2026), this opens the door to exciting new theoretical research, potentially unifying various forms of usability and learnability in a single mathematical framework.

4.2 AI DISCLOSURE

We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative

and thematic data analysis, or interpret results. We did not use AI tools to generate synthetic data sets or assist with translation. We did use generative AI tools in a limited capacity to create or edit code and identify relevant literature. We have reviewed all AI-assisted work. LLM-generated code and AI auto-completion tools were verified by at least 1 author. Use of generated code was limited to smaller helper pieces of code (smart autocomplete). Literature identified by LLMs were reviewed for relevance by at least 1 author. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

4.3 Reproducibility statement

All neural network models used in our experiments are pre-trained and from public repositories, available through PyTorch found at `https://docs.pytorch.org/vision/main/models.html` (Ansel et al., 2024). Source code for our stitching and compatibility experiments are publicly available on GitHub.[2]

4.4 Ethics statement

No direct risks were taken in the production of this publication. One of our motivations for this work is improving neural prosthetics, however, our current conclusions do not have any negative or positive impact on interested parties. No neural networks were harmed in the process of this research.

References

Bardia Abbasi and Joseph F. Rizzo. Advances in Neuroscience, Not Devices, Will Determine the Effectiveness of Visual Prostheses. *Seminars in Ophthalmology*, 36(4):168–175, May 2021. ISSN 0882-0538. doi: 10.1080/08820538.2021.1887902.

Antonio Almudévar and Alfonso Ortega. Bridging functional and representational similarity via usable information, 2026. URL `https://arxiv.org/abs/2601.21568`.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, pp. 929–947, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366. URL `https://doi.org/10.1145/3620665.3640366`.

Ben Baker, Richard D. Lange, Andrew Richmond, Nikolaus Kriegeskorte, Rosa Cao, Xaq Pitkow, and Odelia Schwartz. Use and Usability: Concepts of representation in philosophy, neuroscience, cognitive science, and computer science. *NBDT*, 2026.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 225–236. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/01ded4259d101feb739b06c399e9cd9c-Paper.pdf`.

Enric Boix-Adsera, Hannah Lawrence, George Stepaniants, and Philippe Rigollet. Gulp: a prediction-based metric between representations. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 7115–7127. Curran Associates, Inc., 2022. doi: 10.52202/

[2]URL not provided during double-blind review.

068431-0516. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/2f0435cffef91068ced08d7c7d8e643e-Paper-Conference.pdf.

Lukas Braun, Erin Grant, and Andrew M Saxe. Not all solutions are created equal: An analytical dissociation of functional and representational similarity in deep linear neural networks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 5355–5382. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/braun25a.html.

Nathan Cloos, Moufan Li, Markus Siegel, Scott Brincat, Earl Miller, Guangyu Robert Yang, and Christopher Cueva. Differentiable optimization of similarity scores between models and brains. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 63438–63457, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/9edf04e9b128ed5085189205fc94e21f-Paper-Conference.pdf.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.*, 13(null):795–828, March 2012. ISSN 1532-4435.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 5656–5668. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2cb274e6ce940f47beb8011d8ecb1462-Paper.pdf.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, June 2009. ISBN 978-1-4244-3992-8. doi: 10.1109/CVPR.2009.5206848.

Jörn Diedrichsen and Nikolaus Kriegeskorte. Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLOS Computational Biology*, 13(4):1–33, 04 2017. doi: 10.1371/journal.pcbi.1005508. URL https://doi.org/10.1371/journal.pcbi.1005508.

Shimon Edelman. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(4):449–467, August 1998. ISSN 0140-525X, 1469-1825. doi: 10.1017/S0140525X98001253. URL https://www.cambridge.org/core/product/identifier/S0140525X98001253/type/journal_article.

Eduardo Fernández and Richard A. Normann. CORTIVIS Approach for an Intracortical Visual Prostheses. In Veit Peter Gabel (ed.), *Artificial Vision: A Clinical Guide*, pp. 191–201. Springer International Publishing, Cham, 2017. ISBN 978-3-319-41876-6. doi: 10.1007/978-3-319-41876-6_15.

Marc Finzi, Shikai Qiu, Yiding Jiang, Pavel Izmailov, J. Zico Kolter, and Andrew Gordon Wilson. From Entropy to Epiplexity: Rethinking Information for Computationally Bounded Intelligence, March 2026.

Patricia Grant, Lindsey Spencer, Aimee Arnoldussen, Rich Hogle, Amy Nau, Janet Szlyk, Jonathan Nussdorf, Donald C. Fletcher, Keith Gordon, and William Seiple. The functional performance of the brainport v100 device in persons who are profoundly blind. *Journal of Visual Impairment & Blindness*, 110(2):77–88, 2016. doi: 10.1177/0145482X1611000202. URL https://doi.org/10.1177/0145482X1611000202.

David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12):2639–2664, December 2004. ISSN 0899-7667. doi: 10.1162/0899766042321814. URL https://doi.org/10.1162/0899766042321814. _eprint: https://direct.mit.edu/neco/article-pdf/16/12/2639/815992/0899766042321814.pdf.

Sarah E. Harvey, Brett W. Larsen, and Alex H. Williams. Duality of bures and shape distances with implications for comparing neural representations. In Marco Fumero, Emanuele Rodolá, Clementine Domine, Francesco Locatello, Karolina Dziugaite, and Caron Mathilde (eds.), *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, volume 243 of *Proceedings of Machine Learning Research*, pp. 11–26. PMLR, 15 Dec 2024a. URL https://proceedings.mlr.press/v243/harvey24a.html.

Sarah E. Harvey, David Lipshutz, and Alex H. Williams. What Representational Similarity Measures Imply about Decodable Information. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, October 2024b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Adriano Hernandez, Rumen Dangovski, Peter Y. Lu, and Marin Soljacic. Model stitching: Looking for functional similarity between representations, 2023. URL https://arxiv.org/abs/2303.11277.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, January 2017.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of Neural Network Models: A Survey of Functional and Representational Measures. *ACM Computing Surveys*, 57(9):1–52, September 2025. ISSN 0360-0300, 1557-7341. doi: 10.1145/3728458.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Nikolaus Kriegeskorte. Relating population-code representations between man, monkey, and computational models. *Frontiers in Neuroscience*, 3(3):363–373, 2009. ISSN 16624548. doi: 10.3389/neuro.01.035.2009.

Nikolaus Kriegeskorte and Jörn Diedrichsen. Peeling the onion of brain representations. *Annual Review of Neuroscience*, 42(Volume 42, 2019):407–432, 2019. ISSN 1545-4126. doi: https://doi.org/10.1146/annurev-neuro-080317-061906. URL https://www.annualreviews.org/content/journals/10.1146/annurev-neuro-080317-061906.

Richard D Lange, Devin Kwok, Jordan Kyle Matelsky, Xinyue Wang, David Rolnick, and Konrad Kording. Neural networks as paths through the space of representations, 2023. URL https://openreview.net/forum?id=wAQU0Frxoa.

Karel Lenc and Andrea Vedaldi. Understanding Image Representations by Measuring Their Equivariance and Equivalence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 991–999, June 2015.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/a7a3d70c6d17a73140918996d03c014f-Paper.pdf.

John Morrison. Algorithms for Neural Networks, August 2024. URL https://2024.ccneuro.org/pdf/171_Paper_authored_CCN-abstract-(with-name).pdf. Unpublished manuscript for poster presented at Cognitive Computational Neurosience (CCN) 2024.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, January 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling gobal representations for visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. doi: 10.1109/ICCV48922.2021.00042.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability, November 2017.

David Rolnick and Konrad P. Kording. Reverse-engineering deep relu networks, 2020. URL https://arxiv.org/abs/1910.00744.

Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. ModelDiff: A framework for comparing learning algorithms. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 30646–30688. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shah23a.html.

Robert K. Shepherd, Mohit N. Shivdasani, David AX Nayagam, Christopher E. Williams, and Peter J. Blamey. Visual prostheses for the blind. *Trends in Biotechnology*, 31(10):562–571, October 2013. ISSN 01677799. doi: 10.1016/j.tibtech.2013.07.001.

Damian Smith, Harvey Mannering, and Antonia Marcu. Functional alignment can mislead: examining model stitching. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Leslie N. Smith. Cyclical Learning Rates for Training Neural Networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472, March 2017. doi: 10.1109/WACV.2017.58.

Gilbert Strang. *Introduction to Linear Algebra*. Cambridge press, Wellesley, 5th edition edition, 2016. ISBN 978-0-9802327-7-6.

Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C. Love, Christopher J. Cueva, Erin Grant, Iris Groen, Jascha Achterberg, Joshua B. Tenenbaum, Katherine M. Collins, Katherine L. Hermann, Kerem Oktar, Klaus Greff, Martin N. Hebart, Nathan Cloos, Nikolaus Kriegeskorte, Nori Jacoby, Qiuyi Zhang, Raja Marjieh, Robert Geirhos, Sherol Chen, Simon Kornblith, Sunayana Rane, Talia Konkle, Thomas P. O'Connell, Thomas Unterthiner, Andrew K. Lampinen, Klaus-Robert Müller, Mariya Toneva, and Thomas L. Griffiths. Getting aligned on representational alignment, 2024. URL https://arxiv.org/abs/2310.13018.

Marvin Theiss, Lukas Braun, Andrew M Saxe, and Erin Grant. Parameter symmetries determine representational geometry in overparameterized nonlinear networks. In *ICML 2026 Workshop on Weight-Space Symmetries: from Foundations to Practical Applications*, 2026. URL https://openreview.net/forum?id=CT8TbdqYmn.

Neil Traft. Bridging large gaps in neural network representations with model stitching. In *UniReps: 3rd Edition of the Workshop on Unifying Representations in Neural Models*, 2025. URL https://openreview.net/forum?id=e4wKQHvsUi.

Chuanqing Wang, Di Wu, Chaoming Fang, Jie Yang, and Mohamad Sawan. Exploring effective stimulus encoding via vision system modeling for visual prostheses. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, pp. 15384–15396, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/433f75a8dd75fa81f38e19efcfd2fc0e-Paper-Conference.pdf.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4738–4750. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/252a3dbaeb32e7690242ad3b556e626b-Paper.pdf.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *CoRR*, abs/2002.10689, 2020. URL https://arxiv.org/abs/2002.10689.

A Appendix

A.1 Details for Figure 2D

To compute quantitative values in Figure 2D supporting the qualitative argument, we simulated the simple 3-dimensional data space and the simple models shown in the other panels of the Figure. Specifically, we set up a 3-dimensional data space using a 3d meshgrid on [0,1,2,3] for s_1, s_2, and s_3. To induce a positive correlation between s_1 and s_2, we dropped all values where $s_1 = 0 \wedge s_2 = 3$ and where $s_1 = 3 \wedge s_2 = 0$. We repeated this to induce correlation between s_1 and s_3. This resulted in a total of 50 unique (s_1, s_2, s_3) triples to serve as input data, which we treated as uniformly distributed. We then centered each of s_1, s_2, and s_3 so that biases would be unnecessary in the linear models. We defined labels y* by the sign of the centered s_1.

We defined three two-layer linear models:

$$A(\boldsymbol{s}) = [1 \quad 0] \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \boldsymbol{s}$$

$$B(\boldsymbol{s}) = [0 \quad 1] \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{s}$$

$$C(\boldsymbol{s}) = [1 \quad 0] \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{s}$$

as well as the hybrid (stitched) models

$$AC(\boldsymbol{s}) = [1 \quad 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \boldsymbol{s}$$

$$BC(\boldsymbol{s}) = [1 \quad 0] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{s}\,,$$

where the middle identity matrices express the stitching transformation.

From these models and stimuli we directly calculated Linear CKA (Kornblith et al., 2019) on the two-dimensional hidden spaces of the original models, and we computed performance in terms of classification accuracy of the hybrid models. These are the values reported in Figure 2.

A.2 Proofs for Distinguishing activity correlation from stitching

Definition A.2.1 (Rank-order violation)**.** *A rank-order violation exists between two measurements $f : O \to \mathbb{R}$ and $g : O \to \mathbb{R}$ if there exists three objects A, B, and $C \in O$ such that:*

$$f(A, C) < f(B, C) \text{ and } g(A, C) > g(B, C)$$

Definition A.2.2 (Activity-correlation between layers and activity-correlation methods)**.** *We have two neural models A and B. A pair of neural layers (or blocks) ℓ_A and ℓ_B are activity correlated if there is are vectors of neural activation n_i^A from ℓ_A and n_i^C from ℓ_B with outputs that correlate across stimulus from distribution p (they are viewed as playing similar roles as components of a representation). The higher correlation between neurons n_i^A and n_i^B and the higher number of pairs of correlated neurons n_i^A and n_i^B implies a stronger activity-correlation. An activity-correlation method uses activity-correlation as a component of quantifying it's measure.*

Definition A.2.3 (Neural Stitching)**.** *Given an arbitrary neural model M that can be organized into a sequence of layers (or blocks) $M = (l_0, l_1, ..., l_i, ...l_z)$ where l_0 is the input layer, l_z is the output layer, and l_i is only connected to l_{i-1} then each layer (or block) is equivalently a function $l_i : \mathbb{R}^{i-1} \to \mathbb{R}^i$ where $\mathbb{R}^i$ is the dimensionality of layer l_i's output or otherwise described as the composition of functions corresponding to its own layers (or blocks)*

$$z_M = \ell_z \circ ... \circ \ell_l \circ ... \circ \ell_1 \circ \ell_0$$

We can separate the original model M at the layer ℓ into the upstream sub-model $M^{\ell}_{\wedge} : \mathcal{S} \to \mathcal{X}^{\ell}$ and a downstream sub-model $M^{\ell}_{\vee} : \mathcal{X}^{\ell} \to \mathcal{Y}$ such that the original model is expressible as function composition $M = M^{\ell}_{\vee} \circ M^{\ell}_{\wedge}$.

*Let A and B be neural models like M where ℓ_A is a layer of A and ℓ_B is a layer from B. We say that ℓ_A is **stitchable** into ℓ_B to the extent that a hybrid model,*

$$A^{\ell_A} \times B^{\ell_B} = B^{\ell_B}_{\vee} \circ T \circ A^{\ell_A}_{\wedge},$$

*comprising the upstream part of A, a "stitching layer" T selected from a family of transforms $T \in \mathcal{T}$, and the downstream part of B, performs well on a relevant benchmark, optimized only with respect to the parameters of T (Lenc & Vedaldi, 2015; Bansal et al., 2021; Csiszárik et al., 2021). We then define **stitchability** by the best-case loss*

$$\textit{stitchability}(p, A^{\ell_A}_{\wedge}, \mathcal{T}, B^{\ell_B}_{\vee}, \text{loss}) = -\min_{T \in \mathcal{T}} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p} \left[\text{loss}(B^{\ell_B}_{\vee}(T(A^{\ell_A}_{\wedge}(\mathbf{x}))), \mathbf{y})\right] . \qquad \text{(equation 2 restated)}$$

Loss is negated so that larger numbers correspond to "greater stitchability."

Theorem A.2.4. *Rank-order violations exist between activity-correlation methods and stitching with affine stitching layers.*

Proof. This is a simple proof by construction. For the sake of simplicity we choose CKA as representative example of activity-correlation methods for use in the proof and referrer to it as "correlation".

$$\text{correlation}(p, A^{\ell_A}_{\wedge}, B^{\ell_B}_{\wedge}) = \frac{\|\mathbf{\Sigma}_{AB}\|^2_{\text{F}}}{\|\mathbf{\Sigma}_{AA}\|_{\text{F}} \|\mathbf{\Sigma}_{BB}\|_{\text{F}}} , \qquad \text{(equation 1 restated)}$$

where $\mathbf{\Sigma}_{AB}$ is the cross-covariance between features from ℓ_A and features from ℓ_B and $\| \cdot \|_{\text{F}}$ is the Frobenius norm Cortes et al. (2012); Kornblith et al. (2019).

Suppose we are given three models, A, B, and C, each of which is performing a binary classification of x given some data. The data consist of objects with three binary features $S_i = \{s_1, s_2, s_3\}$, and we will assume that each model explicitly represents a subset of these features in a hidden layer. Additionally, the correct binary classification of x is identical to the value of s_1, thus any model that outputs the value of s_1 will have perfect performance.

We construct the three neural models such that s_1 will be a crucial feature for A. Model A relies solely on the feature s_1 to decide whether x belongs in the group and only contains a representation of s_1 in its hidden layer. Model B uses all features in S_i except s_0 to make its decision, model B only represents s_2 and s_3, using s_2 to classify x. Model C's hidden layer contains representations of features s_2 and s_3, like model B, but its output is decided by s_3. The constructions of A, B, C are illustrated in Figure 6A. Let distribution of x be such that the covariances of s_1, s_2, and s_3 are $\text{cov}(s_1, s_1) = \frac{1}{2}$, $\text{cov}(s_2, s_2) = \frac{1}{2}$, $\text{cov}(s_3, s_3) = \frac{1}{2}$, $\text{cov}(s_1, s_2) = \frac{1}{2}$, $\text{cov}(s_1, s_3) = 0$, and $\text{cov}(s_3, s_3) = 1$.

Lemma A.2.5.

$$\textit{correlation}(p, A^{\ell_A}_{\wedge}, C^{\ell_C}_{\wedge}) < \textit{correlation}(p, B^{\ell_B}_{\wedge}, C^{\ell_C}_{\wedge}) \qquad (5)$$

Proof. The cross-covariance matrices $\|\mathbf{\Sigma}_{AA}\|$, $\|\mathbf{\Sigma}_{BB}\|$, $\|\mathbf{\Sigma}_{CC}\|$, $\|\mathbf{\Sigma}_{AC}\|$, and $\|\mathbf{\Sigma}_{BC}\|$ are derived from the table. The Frobenius norm of each cross-covariance matrix are $\|\mathbf{\Sigma}_{AA}\|_{\text{F}} = 1$, $\|\mathbf{\Sigma}_{BB}\|_{\text{F}} = 1$, $\|\mathbf{\Sigma}_{CC}\|_{\text{F}} = 1$, $\|\mathbf{\Sigma}_{AC}\|_{\text{F}} = \sqrt{\frac{1}{2}}$, and $\|\mathbf{\Sigma}_{BC}\|_{\text{F}} = \sqrt{\frac{3}{2}}$. Therefore,

$$\text{correlation}(p, A^{\ell_A}_{\wedge}, C^{\ell_C}_{\wedge}) = \frac{1}{2}$$

$$\text{correlation}(p, B^{\ell_B}_{\wedge}, C^{\ell_C}_{\wedge}) = \frac{3}{2}$$

□

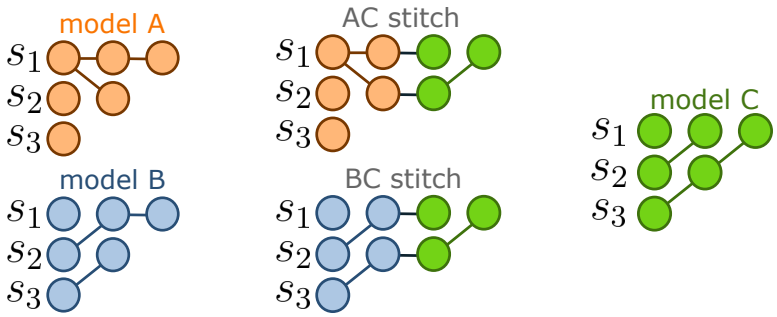

Figure 5: Illustrations of model A,B, and C as well as hybrid model AC made from stitching A into C and BC made from stitching B into C.

Lemma A.2.6.

$$\textit{stitchability}(p, A_{\wedge}^{A_r}, \mathcal{T}, C_{\vee}^{C_r}, \text{loss}) > \textit{stitchability}(p, B_{\wedge}^{B_r}, \mathcal{T}, C_{\vee}^{C_r}, \text{loss})$$

Proof. Notice there exists a mapping T_A^* between the hidden layer A_r and the C_r layer such that $C_{\vee}^{C_r} \circ T_A^* \circ A_{\wedge}^{A_r}(x) = s_1$ for all $x \in p$ that being mapping the s_1 of hidden layer A_r to s_3 of hidden layer C_r. Observe that between hidden layer B_r and hidden layer C_r there does not exist a mapping such that $C_{\vee}^{C_r} \circ T \circ B_{\wedge}^{B_r}(x) = s_1$ for all $x \in p$. Since s_1 achieves perfect performance, $\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T_A^*(A_{\wedge}^{A_r}(\mathbf{x}))), \mathbf{y})\right]$ will be less than $\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T(B_{\wedge}^{B_r}(\mathbf{x}))), \mathbf{y})\right]$ for all $T \in \mathcal{T}$. Therefore,

$$-\min_{T\in\mathcal{T}} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T(A_{\wedge}^{A_r}(\mathbf{x}))), \mathbf{y})\right] > -\min_{T\in\mathcal{T}} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T(B_{\wedge}^{B_r}(\mathbf{x}))), \mathbf{y})\right]$$

$$\text{stitchability}(p, A_{\wedge}^{A_r}, \mathcal{T}, C_{\vee}^{C_r}, \text{loss}) > \text{stitchability}(p, B_{\wedge}^{B_r}, \mathcal{T}, C_{\vee}^{C_r}, \text{loss})$$

□

Together **Lemma A.2.5** and **Lemma A.2.6** obtain a rank order violation following **Definition A.2.1**, concluding the proof of Theorem A.2.4.

□

We conjecture the rank order violation shown above generalizes to other activity correlation methods, such as representational similarity analysis which has shown to be separable from downstream function (Theiss et al., 2026), as long as those methods can separate function from activity correlation and other contributing quantities can be held constant between two models being compared.

A.3 Proofs for Distinguishing Stitching from Compatibility

Another rank order violation can also be shown between stitchability and compatibility with a construction. The construction capitalizes on the stitching layers limitation to linear decoding. Internal representations that are highly usable by the downstream component, but are formatted in a nonlinear embedding will appear like useless noise even when transformed by a well-trained stitching layer. Despite its nonlinearity, if it is usable then the downstream component would learn how to exploit those representations.

Theorem A.3.1. *Rank-order violations exist between stitching with affine stitching layers and compatibility*

Proof. Suppose we are given three feed-forward neural networks A, B, and C with rectified linear activation functions. All three models have the same binary classification task to identify two different objects $O = o_-, o_+$. Let ℓ_A, ℓ_B, and ℓ_C be layers from the respective models. Models A

and B are selected to be upstream donors for model C. The upstream component at ℓ_A provided by A will be called $A^{\ell_A}_{\wedge}$, the upstream component at ℓ_B from B will be called $B^{\ell_B}_{\wedge}$, and $C^{\ell_C}_{\vee}$ is the downstream component from C at ℓ_C. C is explicitly constructed as shown in Figure 6.

All three components represent the objects using two scalar features r_x and r_y, together those features form a two dimensional representation space $\mathbb{R}_{xy}$. The upstream component $A^{\ell_A}_{\wedge}$ provides XOR embedded representations of the objects in O that lead to the outputs visualized in Figure 7. The embedding is labeled as A_υ. The upstream component $B^{\ell_B}_{\wedge}$ embeds objects of O in $\mathbb{R}_{xy}$ using a partial linear embedding, labeled B_υ and illustrated by it's outputs shown in 7. The upstream components' structure are arbitrarily constructed to get the outputs visualized. One should note that the crucial property of model $B^{\ell_B}_{\wedge}$'s embedding is that it is partially linear, but also taken as a whole is a much more complex nonlinear embedding than model $A^{\ell_A}_{\wedge}$'s XOR embedding.

Notice lemma A.3.2, indicates that both A_υ and B_υ maintain the same mutual information from the same input.

Lemma A.3.2. *There exists an invertible function $h : \mathbb{R}_{xy} \to \mathbb{R}_{xy}$ between A_υ and B_υ by mapping quadrants.*

Proof. We begin by dividing the embedding spaces of A_υ and B_υ into quadrants and sub-quadrants, as depicted in Figure 7, then map all sub-quadrants from the embedding space of A_υ to the embedding space of B_υ. The first quadrants are created by drawing two lines parallel to the y-axis that intersect with the positive side of the x-axis, the first line will be called υ_{u1} and the second line υ_{u2} intersection will be strictly greater than υ_{u1}. Another two vertical lines will be drawn on the negative side of the x-axis, with υ_{ψ_1} being closer to the origin than υ_{ψ_2}.

h is constructed to map all points in Quadrant I and Quadrant IV to themselves. Split quadrant II into Q_{IIa}, all the points between the y-axis and line υ_{u_1}, and Q_{IIb}, all the points with larger values of x of line υ_{u_1}. The split forms two subsets of space around υ_{u_1}. Apply the same process instead using line $\upsilon_{\psi 1}$ to form Q_{IIIa} and Q_{IIIb}. h will uniquely map each points in Q_{IIa} to each point in Q_{IIIa} and map each point in Q_{IIIa} to each unique point in Q_{IIa}.

We will draw two more lines to for h's construction, υ_{k_1} and υ_{k_2}, both lines are parallel to the x-axis, reside on the negative side of the y-axis, and υ_{k_2} has a y-intercept less than υ_{k_1}'s intercept. We next divide Q_{IIb} into two subsets of space, $Q_{IIb'}$ and Q_{IIb_c}, using line υ_{k_2} and υ_{u_2}. Let Q_{IIb_c} be all the points with more positive x values than υ_{u_2} and more negative y values than υ_{k_2}, define $Q_{IIb'}$ as the rest of the points in Q_{IIb}. We construct h such that all points in $Q_{IIb'}$ are mapped to themselves. Q_{IIIb} will also be separated into two subsets $Q_{IIIb'}$ and Q_{IIIb_c}. Q_{IIIb_c} we be the set of all points with x values less than υ_{ψ_2} and y values greater than υ_{k_1}. $Q_{IIIb'}$ will map to all the points in Q_{IIIb} not found in $Q_{IIIb_{xtr}}$. h is constructed to map all points in $Q_{IIIb'}$ to themselves. h maps each point in Q_{IIIb_c} to a unique point in Q_{IIb_c}, h applies the same unique mapping on $Q_{IIb_{xtr}}$ to $Q_{IIIb_{xtr}}$. We end the construction with all quadrants of $\mathbb{R}_{xy}$ of embedding space A_υ being mapped by h to embedding space of B_υ.

The function h, following its construction, is a transformation from embedding A_υ to B_υ. Moreover, h is a injective because the construction stipulates a unique mapping for each point in every quadrant. h is also surjective as the unity of every subset of space h maps to is the entire space of $\mathbb{R}_{xy}$. Thus, h is an invertible function between embeddings A_υ and B_υ. □

Now we turn our attention to the downstream component of C and the stitching layer. Consider the observation and the lemmas below:

Observe that $C(x, y) = W^T \begin{pmatrix} x \\ y \end{pmatrix}$ by its construction.

$$C(x,y) = 0 * reLu(x-y) - W_{xy} * reLu(-x-y) + W_{xy} * reLu(x+y) + 0 * reLu(y-x)$$

$$C(x,y) = W_{xy} * reLu(x+y) - W_{xy} * reLu(-x-y)$$

$$C(x,y) = W_{xy}(x+y) = W_{xy}x + W_{xy}y = \begin{pmatrix} w_{xy} \\ 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = W^T \begin{pmatrix} x \\ y \end{pmatrix}$$

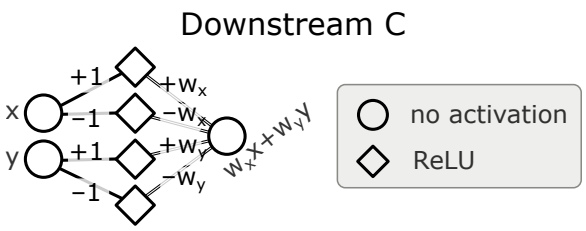

Figure 6: Architecture and initial weights of the downstream model $C_\vee$. Note that $C_\vee(\mathbf{x}) = [\boldsymbol{w}_x\ \boldsymbol{w}_y]\mathbf{x}$, so the downstream model implements a linear readout of its inputs despite the presence of ReLU nonlinearities in its middle layer. This is the same as Figure 3B, except there we set $\boldsymbol{w} = \pm 1$ for all weights for simplicity; here we give the weights names to aid in the proof.

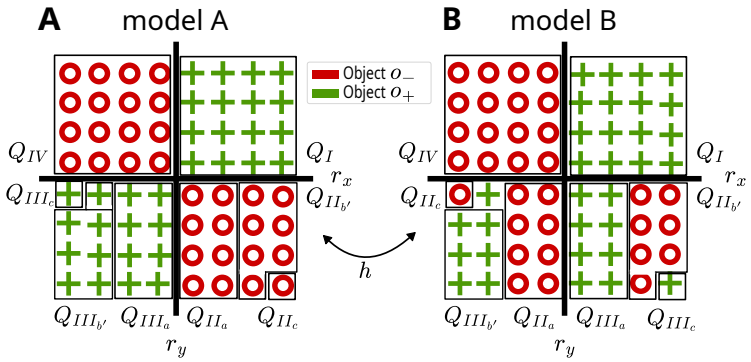

Figure 7: Visualizations of the outputs of $A^{\ell_A}_\wedge$ **(A)** and $B^{\ell_B}_\wedge$ **(B)**. Notice that the function h is a bijective map from **A)** to **B)** by swapping around the labeled quadrants. **A)** The output of $A^{\ell_A}_\wedge$ in representation space $\mathbb{R}_{xy}$ also called the embedding A_υ. **B)** The output of $B^{\ell_B}_\wedge$ in representation space $\mathbb{R}_{xy}$ also called the embedding B_υ.

Therefore, C is linear.

Lemma A.3.3. *Assume a linear stitching layer $T(h^{l_a}) = W_s g^\ell$ be the optimal linear decoder of y from g^ℓ according to $loss(\cdot, y)$. Now, if the downstream model C is linear i.e. $\exists A(g^\ell) = Ag^\ell$, then the best stitching performance can do no better than the best linear decoder,*

$$loss(C(T(g^\ell)) \geq loss(W_{opt} g^\ell, y)$$

with equality whenever the column span of W_{opt} is contained in the column span of A, i.e $Col(W_{opt} \subset Col(A))$

Proof. Both C and the stitching layer are linear, so their composition is linear, equal to $AW_T h^\ell$, by assumption, the best linear decoder of y from h^ℓ is given by $W_{opt} h^\ell$, hence the inequality. Equality is achieved if there exists a W_T such that $AW_T = W_{opt}$. The fundamental theorem of linear algebra –Strang (2016)– states that a solution (for each column of W_opt) exists if and only if they are contained in the span of the columns of A. □

Lemma A.3.4.

$$\textit{stitchability}(A^{\ell_A}_\wedge, C^{\ell_C}_\vee) < \textit{stitchability}(B^{\ell_B}_\wedge, C^{\ell_C}_\vee)$$

Proof. Our choice of stitching family $\mathcal{T}$ is restricted to the set of affine transformations. Knowing that both C and all $T \in T$ are linear means no choice of T performs no better than the optimal linear decoder for C because of lemma A.3.3. Furthermore, we construct the stitching family such that $AW_T = W^A_{\text{opt}}$ where $W_{T^*_A}$ is the stitching layer transformation for the $A^{\ell_A}_\wedge$ and $BW_{T^*_B} = W^B_{\text{opt}}$ as the optimal decoder for $B^{\ell_B}_\wedge$, i.e. both model A and model B have the best stitching possible. We train each stitching layers enough time to find their respective optimal transformations $T^*_A = W_{T^*_A}$ and $T^*_B = W_{T^*_B}$ for their own upstream components. Thus, the hybrid models $A^{\ell_A} \times C^{\ell_C} =$

$C_{\vee}^{\ell_C} \circ T_A^* \circ A_{\wedge}^{\ell_A}$'s and $A^{\ell_A} \times C^{\ell_C} = C_{\vee}^{\ell_C} \circ T_B^* \circ B_{\wedge}^{\ell_B}$'s performance is equal to the performance of their optimal linear decoders on the binary classification task.

Observing A_v (Figure 7A) geometrically, it is clear that observations of objects O_- are strictly confined to points in quadrant IV and II while O_+ observations are restricted to quadrant I and III. Therefore decoder success is dependent on how well they can separate quadrant IV and II from quadrant I and III. A linear decoder could at most separate quadrant II from quadrant I and quadrant IV from quadrant III or separate quadrant I from quadrant IV and quadrant III from quadrant II, using the x-axis. Any other line through the origin would have equivalent separation for the embedding space of A_v. The optimal linear decoder correctly classifies at most 2 quadrants in the embedding space, one quadrant from each class of object. The linear decoder is only capable of correctly classifying 50% of o_- objects and 50% O_+ objects. Thus the hybrid model $A^{\ell_A} \times C^{\ell_C}$'s best performance will incorrectly classify objects O_- and O_+ only 50% of the time.

In comparison, we can see geometrically for B_v (Figure 7B) that the there exists a line going vertically through origin which separates 75% of O_- observations and 75% of the O_+ observations. The optimal linear decoder B_v will at least correctly classify 75% of each object class correctly, meaning the the hybrid model $B^{\ell_B} \times C^{\ell_C}$ best performance will incorrectly classify at most a quarter of their object o_- and a quarter of the object o_+.

Therefore, any reasonably defined loss function for a binary classification task would show: $\mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{\ell_C}(T_A^*(A_{\wedge}^{\ell_A}(\mathbf{x}))), \mathbf{y})\right] < \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{\ell_C}(T_B^*(B_{\wedge}^{\ell_B}(\mathbf{x}))), \mathbf{y})\right]$ since these are both optimal with regards to the task then

$$-\min_{T\in\mathcal{T}} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T(A_{\wedge}^{A_r}(\mathbf{x}))), \mathbf{y})\right] < -\min_{T\in\mathcal{T}} \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p}\left[\text{loss}(C_{\vee}^{C_r}(T(B_{\wedge}^{B_r}(\mathbf{x}))), \mathbf{y})\right]$$

$$\text{stitchability}(A_{\wedge}^{\ell_A}, C_{\vee}^{\ell_C}) < \text{stitchability}(B_{\wedge}^{\ell_B}, C_{\vee}^{\ell_C})$$

□

A.3.4 shows that $A_{\wedge}^{\ell_A}$ has less stitchability to $C_{\vee}^{\ell_C}$ than $B_{\wedge}^{\ell_B}$. However, suppose we also allow C to learn before measuring performance of the hybrid models, as in the case of compatibility. If $C_{\vee}^{\ell_C}$ is allowed to learn, it is no longer limited to being linear, removing the optimal decoder bound on hybrid model performance.

Theorem A.3.5. *for each neuron in a reLU neural network exists a linear decision boundary (Rolnick & Kording, 2020)*

Lemma A.3.6.

$$\textit{compatibility}(A_{\wedge}^{\ell_A}, C_{\vee}^{\ell_C}) > \textit{compatibility}(B_{\wedge}^{\ell_B}, C_{\vee}^{\ell_C})$$

Proof. As established in Rolnick & Kording (2020), for each neuron in a reLU neural network exists a linear decision boundary, represented by a line across the feature space projected in two dimensions. Our downstream component $C_{\vee}^{\ell_C}$ is a sub-network and thus is itself a neural network being given inputs from the stitching layer. The feature space inputted into $C_{\vee}^{\ell_C}$ is two-dimensional, for both upstream components $A_{\wedge}^{\ell_A}$ and $B_{\wedge}^{\ell_B}$, thus the 2D projections of feature spaces given from the upstream are the same as the feature spaces themselves. C has at most 7 neurons (if we include input from the stitching layer). Therefore, downstream component $C_{\vee}^{\ell_C}$ could be understood as a combination of 7 linear decision boundaries applied to the given upstream feature space. We can use this perspective to set an upper bound for $C_{\vee}^{\ell_C}$ performance with respect to a given upstream component and stitching layer, showing how many observations with different subclasses can be separated by 7 decision boundaries.

First, we consider the feature space given by $A_{\wedge}^{\ell_A}$ or the A_v embedding (Figure 7A). In A_v all observations of different classes can be separated with at least two lines, one vertical line and one horizontal line near the origin. C would be capable of learning the entirety of A_v especially given access to more linear boundaries. Therefore, there exists a learning rate loss^*, and learning algorithm Λ^* for $C_{\vee}^{\ell_C}$ such that the hybrid model $A^{\ell_A} \times C^{\ell_C}$ reaches perfect performance within σ^* budget.

Second, we look at B_v (Figure 7B) the feature space given by $B^{\ell_B}_{\wedge}$. We find that separating out all the classes is not possible with 7 linear decision boundaries. First the observations located in $Q_{IIb_{xtr}}$ would require at least two linear boundaries to separate observations of o_+ from neighboring observations of o_-. $Q_{IIIb_{xtr}}$ would require at least two linear boundaries as well to instead separate o_- observations from o_+. So far B_v requires at least 4 neurons. There would need to be a vertical line to separate out quadrant I and quadrant IV as they have observations of different objects. There would also need to be a horizontal line near the origin to separate quadrant I from Q_{IIa} and separate quadrant IV from Q_{IIIa}. Even with all of these linear boundaries Q_{IIIa} needs to separated from $Q_{IIIb'}$ and Q_{IIa} from $Q_{IIb'}$, meaning there would need to be two more lines. B_v needs at least 8 neurons to have perfect performance, thus perfect performance on B_v is not possible for the downstream component C.

It is possible to achieve perfect performance for A_v and not B_v, meaning there exist cases where $A^{\ell_A} \times C^{\ell_C}$ will outperform $B^{\ell_B} \times C^{\ell_C}$. In any one of these cases, if we compare the compatibility between $A^{\ell_A}_{\wedge}$ and $C^{\ell_C}_{\vee}$ to the compatibility between $B^{\ell_B}_{\wedge}$ and $C^{\ell_C}_{\vee}$ with the budget of σ^* the conclusion will be that:

$$\text{compatibility}(p, A^{\ell_A}_{\wedge}, T^*_A, C^{\ell_C}_{\vee}, \text{loss}^*, \Lambda^*, \sigma^*) > \text{compatibility}(p, B^{\ell_B}_{\wedge}, T^*_B, C^{\ell_C}_{\vee}, \text{loss}^*, \Lambda^*, \sigma^*)$$

. □

Therefore, there exists a rank order violation between compatibility and stitching. □

A.4 Additional Minimal Example Details

A.4.1 Minimal Example of Qualitative Distinction Between Stitchability and Compatibility

The upstream components visualized in Figure 3A were implemented as 16 by 16 matrices. A function had been used to generate the model B matrix from the model A matrix. The downstream component C had been implemented as PyTorch module, where the weights of the hidden layer were hard-coded and the weights of the output were randomly initialized. Each layer had a bias initialized to zero. The hidden layer had activation functions and the output layer had reLu activation functions. The stitching layer was restricted to affine linear transformations. Both the stitching layer and the downstream component C were trained on Binary cross-entropy loss with a 10^{-3}-scaled soft penalty to prevent optimization by collapsing the readout vector to 0 and they both had a training budget of 1000 steps. Both the stitching layer and the downstream component C were trained with stochastic gradient descent momentum 0.9 and weight decay of 10^{-6}. Both were trained with a learning rate of 10^{-2} with a decay of 0.95 every 100 steps.

A.5 Additional Experimental Results

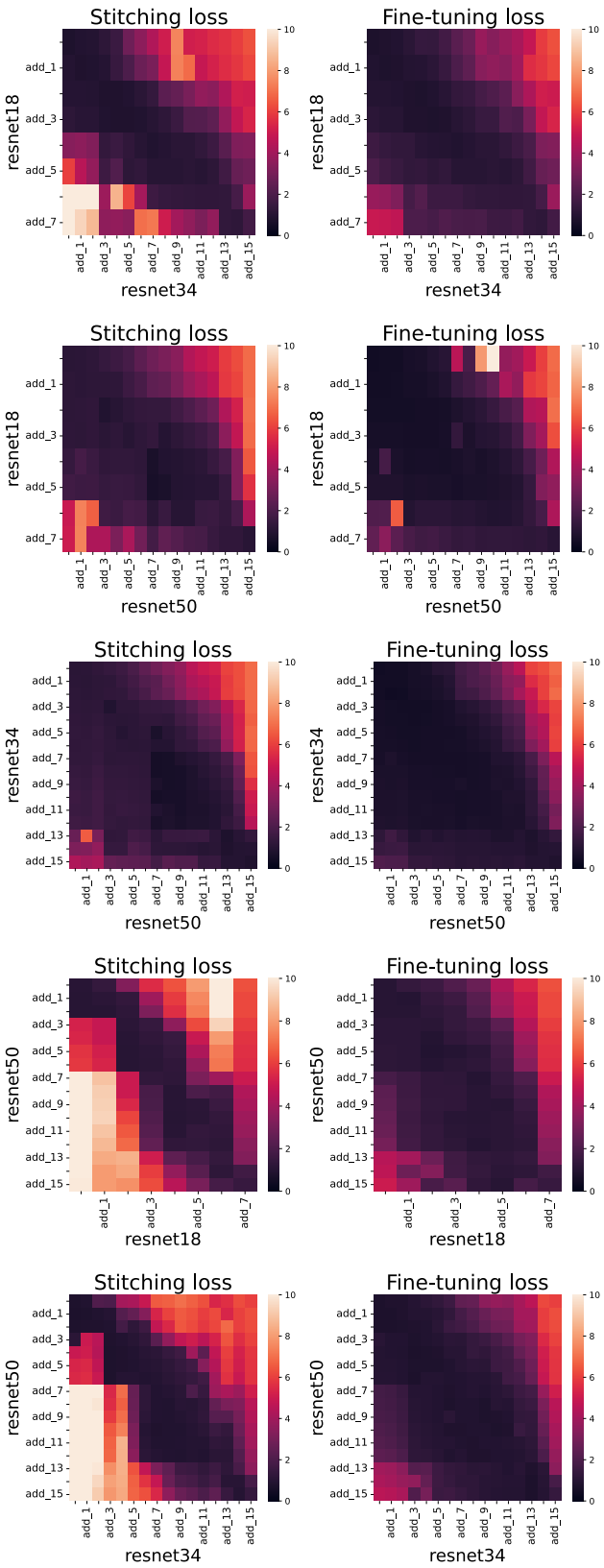

Figure 8: Each Heatmap of the stitching loss and each heatmap of the fine-tuning loss for all given pairs of upstream components and downstream components between each pair of pretrained ResNet models.

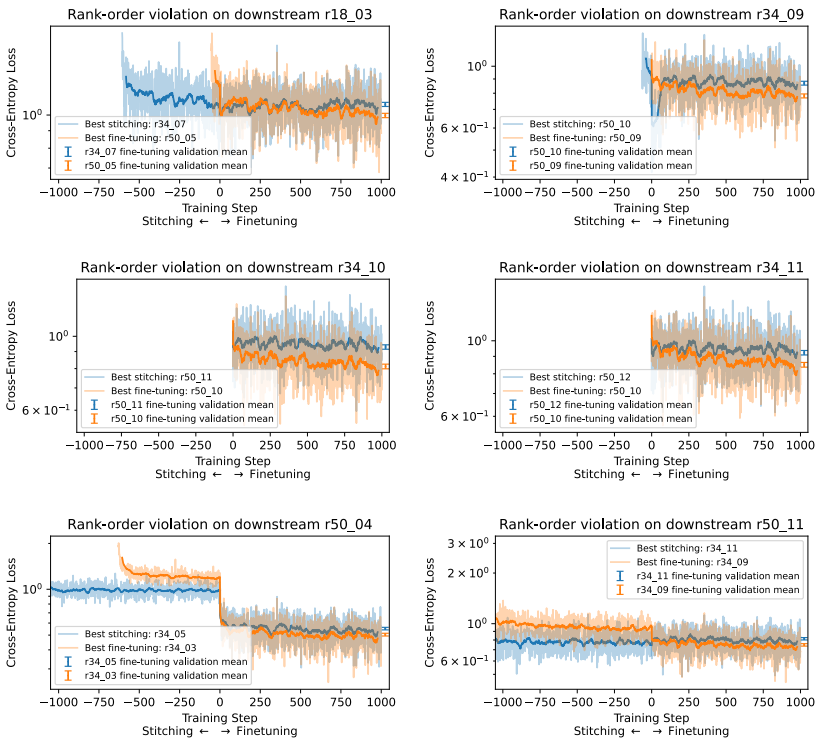

Figure 9: Diagrams of violation between stitching and compatibility. The diagram shows the cross-entropy loss curves for each hybrid model made from the upstream components and the downstream component. The loss curves across the negative steps occurred over the stitching layer training and the loss curves on the positive side occurred over downstream fine-tuning. The true loss curves are made semi-transparent around smoothed average curves for visual clarity. After the loss curves, the diagram displays the validation means of each hybrid models cross entropy loss. For simplicity the names of the upstream and downstream components are shortened into the form "{ResNet-type}_{add-layer}".

Down	BS	BS.S Loss	BS.S Acc	BS.FT Loss	BS.FT Acc	BFT	BFT.S Loss	BFT.S Accuracy	BFT.FT Loss	BFT.FT Acc
r18_00	r50_01	1.032±0.015	73.16%±0.39%	1.024±0.015	73.35%±0.39%	r50_01	1.032±0.015	73.16%±0.39%	1.024±0.015	73.35%±0.39%
r18_01	**r50_02**	1.040±0.015	73.06%±0.39%	1.027±0.015	73.28%±0.39%	**r34_02**	1.140±0.015	70.85%±0.40%	0.979±0.015	74.80%±0.38%
r18_02	r34_04	1.034±0.015	73.52%±0.39%	1.027±0.015	73.73%±0.39%	r50_05	1.045±0.015	73.08%±0.39%	0.937±0.014	75.16%±0.38%
r18_03	**r34_07**	1.079±0.015	72.09%±0.40%	1.074±0.015	72.27%±0.40%	**r50_05**	1.158±0.015	70.25%±0.40%	0.996±0.015	73.77%±0.39%
r18_04	r50_10	1.026±0.015	73.52%±0.39%	1.025±0.015	73.57%±0.39%	r50_12	1.266±0.016	67.73%±0.41%	1.001±0.014	73.49%±0.39%
r18_05	r50_10	1.123±0.015	70.91%±0.40%	1.121±0.015	70.97%±0.40%	r50_09	1.169±0.015	69.92%±0.41%	1.016±0.015	73.30%±0.39%
r18_06	r34_14	1.041±0.016	73.00%±0.39%	1.041±0.016	73.00%±0.39%	r34_14	1.041±0.016	73.00%±0.39%	1.041±0.016	73.00%±0.39%
r18_07	r34_15	1.060±0.012	74.69%±0.38%	1.013±0.013	74.99%±0.38%	r34_15	1.060±0.012	74.69%±0.38%	1.013±0.013	74.99%±0.38%
r34_00	r50_01	0.777±0.013	79.13%±0.36%	0.750±0.013	79.55%±0.36%	r50_01	0.777±0.013	79.13%±0.36%	0.750±0.013	79.55%±0.36%
r34_01	r50_01	0.802±0.013	78.38%±0.36%	0.773±0.013	79.14%±0.36%	r50_01	0.802±0.013	78.38%±0.36%	0.773±0.013	79.14%±0.36%
r34_02	r50_02	0.825±0.014	78.20%±0.36%	0.795±0.013	78.93%±0.36%	r50_02	0.825±0.014	78.20%±0.36%	0.795±0.013	78.93%±0.36%
r34_03	r50_04	0.801±0.013	78.78%±0.36%	0.767±0.013	79.65%±0.36%	r50_04	0.801±0.013	78.78%±0.36%	0.767±0.013	79.65%±0.36%
r34_04	r18_02	0.829±0.013	77.80%±0.37%	0.817±0.013	78.20%±0.36%	r50_05	0.851±0.014	77.09%±0.37%	0.804±0.013	78.27%±0.36%
r34_05	r18_03	0.827±0.013	77.88%±0.37%	0.808±0.013	78.21%±0.36%	r50_05	0.864±0.014	77.07%±0.37%	0.805±0.013	78.59%±0.36%
r34_06	r18_03	0.862±0.014	77.14%±0.37%	0.852±0.014	77.37%±0.37%	r50_08	1.229±0.017	69.45%±0.41%	0.805±0.013	78.28%±0.36%
r34_07	r50_10	0.845±0.014	77.06%±0.37%	0.837±0.014	77.34%±0.37%	r50_10	0.845±0.014	77.06%±0.37%	0.837±0.014	77.34%±0.37%
r34_08	r50_11	0.864±0.014	76.67%±0.37%	0.858±0.014	76.73%±0.37%	r50_12	0.901±0.014	76.18%±0.38%	0.836±0.014	77.76%±0.37%
r34_09	**r50_10**	0.873±0.014	76.59%±0.37%	0.869±0.014	76.72%±0.37%	**r50_09**	0.969±0.015	74.68%±0.38%	0.782±0.013	78.72%±0.36%
r34_10	**r50_11**	0.940±0.014	75.64%±0.38%	0.928±0.014	75.91%±0.38%	**r50_10**	1.004±0.015	74.09%±0.39%	0.811±0.013	78.41%±0.36%
r34_11	**r50_12**	0.932±0.014	75.53%±0.38%	0.922±0.014	75.86%±0.38%	**r50_10**	1.096±0.015	71.73%±0.40%	0.849±0.014	77.71%±0.37%
r34_12	r50_12	1.107±0.014	71.91%±0.40%	0.973±0.014	74.08%±0.39%	r50_15	2.445±0.017	47.02%±0.44%	0.960±0.015	74.76%±0.38%
r34_13	r50_14	1.091±0.013	74.28%±0.39%	0.761±0.013	79.32%±0.36%	r50_14	1.091±0.013	74.28%±0.39%	0.761±0.013	79.32%±0.36%
r34_14	r50_15	1.001±0.012	75.98%±0.38%	0.991±0.012	76.20%±0.38%	r50_15	1.001±0.012	75.98%±0.38%	0.991±0.012	76.20%±0.38%
r34_15	r50_15	1.444±0.011	73.82%±0.39%	1.444±0.011	73.84%±0.39%	r50_15	1.444±0.011	73.82%±0.39%	1.444±0.011	73.84%±0.39%
r50_00	r34_00	1.120±0.008	88.25%±0.28%	0.454±0.010	86.84%±0.30%	r34_00	1.120±0.008	88.25%±0.28%	0.454±0.010	86.84%±0.30%
r50_01	r34_01	1.127±0.008	87.73%±0.29%	0.474±0.010	86.65%±0.30%	r34_02	1.137±0.008	87.75%±0.29%	0.427±0.010	87.75%±0.29%
r50_02	r34_02	1.174±0.009	86.51%±0.30%	0.465±0.010	86.87%±0.30%	r34_02	1.174±0.009	86.51%±0.30%	0.465±0.010	86.87%±0.30%
r50_03	r34_03	0.878±0.010	84.61%±0.32%	0.479±0.010	86.12%±0.31%	r34_04	1.155±0.009	86.06%±0.31%	0.479±0.010	86.49%±0.30%
r50_04	r34_05	0.982±0.009	84.35%±0.32%	0.551±0.011	84.39%±0.32%	r34_03	1.187±0.009	85.21%±0.31%	0.504±0.010	85.94%±0.31%
r50_05	r18_03	1.026±0.010	83.68%±0.33%	0.582±0.011	83.95%±0.32%	r34_07	1.228±0.010	83.19%±0.33%	0.555±0.011	84.24%±0.32%
r50_06	r34_06	1.144±0.010	83.63%±0.33%	0.603±0.011	83.44%±0.33%	r34_05	1.180±0.010	83.09%±0.33%	0.561±0.011	84.23%±0.32%
r50_07	r34_07	0.555±0.010	85.34%±0.31%	0.555±0.010	85.36%±0.31%	r34_07	0.555±0.010	85.34%±0.31%	0.555±0.010	85.36%±0.31%
r50_08	r34_07	0.607±0.011	84.13%±0.32%	0.595±0.011	83.36%±0.33%	r34_07	0.607±0.011	84.13%±0.32%	0.595±0.011	83.36%±0.33%
r50_09	r34_10	0.642±0.011	82.87%±0.33%	0.645±0.012	82.12%±0.34%	r34_09	0.644±0.011	82.98%±0.33%	0.635±0.011	82.95%±0.33%
r50_10	r34_10	0.744±0.011	81.43%±0.34%	0.684±0.012	81.23%±0.34%	r34_10	0.744±0.011	81.43%±0.34%	0.684±0.012	81.23%±0.34%
r50_11	r34_11	0.780±0.012	80.12%±0.35%	0.813±0.013	77.36%±0.37%	r34_09	0.931±0.012	77.83%±0.37%	0.748±0.013	79.80%±0.35%
r50_12	r34_12	0.901±0.012	77.87%±0.37%	0.756±0.013	79.20%±0.36%	r34_12	0.901±0.012	77.87%±0.37%	0.756±0.013	79.20%±0.36%
r50_13	r34_13	0.806±0.013	78.13%±0.37%	0.805±0.013	78.16%±0.37%	r34_13	0.806±0.013	78.13%±0.37%	0.805±0.013	78.16%±0.37%
r50_14	r34_15	0.951±0.014	75.53%±0.38%	0.950±0.015	75.59%±0.38%	r34_15	0.951±0.014	75.53%±0.38%	0.950±0.015	75.59%±0.38%
r50_15	r34_15	0.839±0.014	77.78%±0.37%	0.839±0.014	77.78%±0.37%	r34_15	0.839±0.014	77.78%±0.37%	0.839±0.014	77.78%±0.37%

Table A.5.1: Full table of stitching vs fine-tuning results. Each row is defined by a particular downstream model ("Down"). "BS" = best upstream model after stitching, chosen by lowest validation loss across all possible upstream models. "BFT" = best upstream model after fine-tuning. All hybrid models underwent all phases of training, so for each upstream model we report its loss and accuracy after the stitching stage ("S Loss" and "S Acc") as well as its loss and accuracy after the fine-tuning phase ("FT Loss" and "FT Acc"). All "±" indicate MCSE across all items in the ImageNet validation set.

Gray rows indicate cases where, for one of the hybrid models, accuracy after finetuning is worse than accuracy after initial stitching, indicating that finetuning may have failed. Yellow rows indicate where the best upstream model after stitching ("BS") is different than the best upstream model after fine-tuning ("BFT"), i.e. a potential rank-order violation between stitching and compatibility. Bolded model names indicate rows where the rank-order violation is significant, quantified by testing both conditions that BS.S Acc $>$ BFT.S Acc but also BS.FT Acc $<$ BFT.FT Acc, all at a $p < 0.01$ level.