



Similarity of Neural Network Models: A Survey of Functional and Representational Measures

MAX KLABUNDE, University of Passau, Passau, Germany

TOBIAS SCHUMACHER, University of Mannheim, Mannheim, Germany and RWTH Aachen University, Aachen, Germany

MARKUS STROHMAIER, University of Mannheim, Mannheim, Germany, GESIS - Leibniz Institute for the Social Sciences, Mannheim, Germany, and Complexity Science Hub Vienna, Wien, Austria

FLORIAN LEMMERICH, University of Passau, Passau, Germany

Measuring similarity of neural networks to understand and improve their behavior has become an issue of great importance and research interest. In this survey, we provide a comprehensive overview of two complementary perspectives of measuring neural network similarity: (i) representational similarity, which considers how *activations* of intermediate layers differ, and (ii) functional similarity, which considers how models differ in their *outputs*. In addition to providing detailed descriptions of existing measures, we summarize and discuss results on the properties of and relationships between these measures, and point to open research problems. We hope our work lays a foundation for more systematic research on the properties and applicability of similarity measures for neural network models.

CCS Concepts: • **General and reference** → **Surveys and overviews; Metrics**; • **Computing methodologies** → **Neural networks**; *Learning latent representations; Supervised learning by classification*;

Additional Key Words and Phrases: Deep learning, representational similarity, functional similarity

ACM Reference Format:

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. 2025. Similarity of Neural Network Models: A Survey of Functional and Representational Measures. *ACM Comput. Surv.* 57, 9, Article 242 (May 2025), 52 pages. <https://doi.org/10.1145/3728458>

1 Introduction

Measures to quantify similarity of neural network models have been widely applied in the literature, usually to understand and improve deep learning systems. Examples include research on learning dynamics [98, 101], effects of width and depth [107], differences between supervised and unsupervised models [52], robustness [64, 104], effects of data and model updates

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Grant No.: 453349072.

Authors' Contact Information: Max Klabunde, University of Passau, Passau, Germany; e-mail: max.klabunde@unipassau.de; Tobias Schumacher, University of Mannheim, Mannheim, Germany and RWTH Aachen University, Aachen, Germany; e-mail: tobias.schumacher@uni-mannheim.de; Markus Strohmaier, University of Mannheim, Mannheim, Germany and GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany and Complexity Science Hub, Vienna, Austria; e-mail: markus.strohmaier@uni-mannheim.de; Florian Lemmerich, University of Passau, Passau, Germany; e-mail: florian.lemmerich@unipassau.de.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2025 Copyright held by the owner/author(s).

ACM 0360-0300/2025/05-ART242

<https://doi.org/10.1145/3728458>

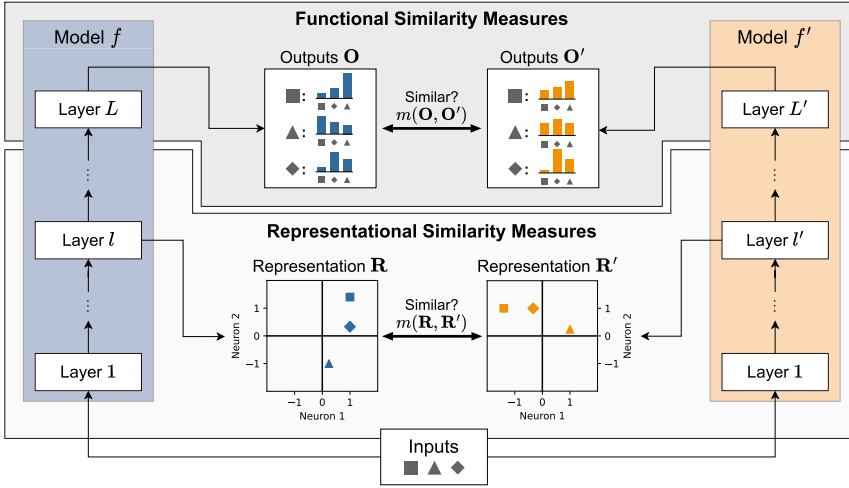


Fig. 1. A conceptual overview of representational and functional similarity. We compare a pair of neural network models f, f' . Functional similarity measures mainly consider the outputs O, O' of the compared models, whereas representational similarity measures consider their intermediate representations R, R' . All models get the same inputs. Specifically in classification tasks, outputs have clear and universal semantics, so that they can be compared in a straightforward manner. In contrast, the geometry of the representations requires more care when measuring their similarity. In the illustration above, for instance, rotating R by 90° would yield an alignment of representations after which they would appear much more similar. Combined, representational and functional measures cover all layers of the models.

[39, 69, 86, 99], evaluating knowledge distillation [134], designing ensembles [163], language representation [53, 54, 75], and generalizability [79, 96, 110].

However, understanding and measuring similarity of neural networks is a complex problem, as there are multiple perspectives on how such models can be similar. In this work, we specifically focus on two key perspectives: *representational* and *functional measures of similarity* (see Figure 1). Representational similarity measures assess how activations of intermediate layers differ, whereas functional similarity measures compare the outputs of neural networks with respect to their task. Both perspectives only provide a partial view on neural network similarity. Seemingly similar representations can still yield different outputs, and, conversely, similar outputs can result from different representations. In that sense, combining these two complementary perspectives provides a more comprehensive approach to analyze similarity between neural networks at all layers.

Given the broad range of research on neural network similarity, numerous representational and functional similarity measures have been proposed and applied, often with lines of research being disconnected from each other. With this work, we provide a comprehensive overview of these two groups of similarity measures that gives a unified perspective on the existing literature and can inform and guide both researchers and practitioners interested in understanding and comparing neural network models.

Measures for representational or functional similarity have been covered in prior work to some extent. Regarding representational similarity, measures for matrix correlation have been reviewed in References [117, 160]. Existing surveys, however, lack coverage of more recent measures or do not consider the context of deep learning. A recent survey by Räuker et al. [120] reviews methods to interpret inner workings of neural networks but discusses representational similarity measures only briefly. Sucholutsky et al. [136] complement this survey by discussing representational

similarity with a focus on bringing together the communities of machine learning, neuroscience, and cognitive science, which have all been working independently on comparing representations. Functional similarity measures have been surveyed in the context of ensemble learning [19, 76], inter-rater agreement [6, 46, 143], model fingerprinting [138], and image and text generation scenarios [18, 22], which each focus on application scenarios with objectives different to our survey. We specifically focus on multiclass classification contexts for functional similarity measures.

To the best of our knowledge, our survey represents the first comprehensive review of representational and functional similarity measures for neural network models. This survey makes the following contributions:

- (1) *Systematic and comprehensive overview*: We formally define the problem of measuring representational and functional similarity in neural networks—the latter in the context of classification—and provide a systematic and comprehensive overview of existing measures.
- (2) *Unified terminology*: We provide detailed definitions, explanations, and categorizations for each measure in a unified manner, facilitating the understanding of commonalities and differences between measures.
- (3) *Analysis of practical properties and applicability*: We discuss the practical properties of existing measures, such as robustness to noise or confounding issues, and connections between existing measures to guide researchers and practitioners in applying these measures.
- (4) *Open research challenges*: We highlight unresolved issues of similarity measures and point out research gaps that can be addressed in the future to improve our understanding of neural networks in general.

While we focus on measures for representational and functional similarity due to their prevalence and general applicability, we acknowledge various other approaches to comparing neural networks. In particular, the measures covered in our survey differ from methods typically used to assess and optimize similarity during model training. We discuss these and other approaches in Appendix E.

2 Similarity of Neural Network Models

We consider the problem of comparing neural networks, which we assume to have the form

$$f = f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(1)}, \quad (1)$$

with each function $f^{(l)} : \mathbb{R}^{D^{(l-1)}} \rightarrow \mathbb{R}^{D^{(l)}}$ denoting a single layer of $D := D^{(l)}$ neurons and a total number of $L \in \mathbb{N}$ layers. These networks operate on a set of N given inputs $\{X_i\}_{i=1}^N$, which we typically assume to be vectors in \mathbb{R}^p , $p \in \mathbb{N}$, although these can also be higher-dimensional structures as occurring in image or video data. We collect these inputs in a matrix $X \in \mathbb{R}^{N \times p}$ so that the i th row X_i corresponds to the i th input. To further simplify notation, we also denote individual inputs X_i as *instances* $i \in \{1, \dots, N\}$. We generally do not make any assumption about the number of features p , the depth of the network L , the width or activation function of any layer $f^{(l)}$, or the training objective.

Similarity of neural network models is then quantified by *similarity measures* m . For simplicity, we also consider measures that quantify *distance* between models as similarity measures, since these concepts are generally equivalent. In our survey, we specifically consider two kinds of similarity, namely *representational similarity* and *functional similarity*. Representational similarity measures consider how the inner activations of neural network models differ, whereas functional similarity measures compare the output behavior of neural networks with respect to a given (classification) task. Combined, these two notions allow for nuanced insights into similarity of neural network models [52, 69, 137].

In the following, we give more thorough definitions of representational and functional similarity. For the rest of this article, we introduce notations for commonly used variables only once. In Appendix A, we provide an overview of notation and several definitions of variables and functions that are used in this article.

2.1 Representational Similarity

Representational similarity measures compare neural networks by measuring similarity between activations of a fixed set of inputs at any pair of layers. Given such inputs X , we define the representation of model f at layer l as a matrix

$$R := R^{(l)} = \left(f^{(l)} \circ f^{(l-1)} \circ \dots \circ f^{(1)} \right) (X) \in \mathbb{R}^{N \times D}. \quad (2)$$

The activations of instance i then correspond to the i th row $R_i = \left(f^{(l)} \circ \dots \circ f^{(1)} \right) (X_i) \in \mathbb{R}^D$, which we denote as *instance representation*. The activations of single neurons over all instances correspond to the columns of R , and we denote the j th column of R as $R_{-,j}$. Like the inputs, we also consider the instance representations R_i to be vectors even though in practice, e.g., in convolutional neural networks, these activations can also be matrices. In such a case, these representations can be flattened (see Appendix B).

Representational similarity measures can be defined as mappings $m : \mathbb{R}^{N \times D} \times \mathbb{R}^{N \times D'} \rightarrow \mathbb{R}$ that assign a similarity score $m(R, R')$ to a pair of representations R, R' , which are derived from different models f, f' but use the same inputs X . While we assume here that representations stem from different models, representational similarity measures can also be used to compare representations of different layers of the same model. Without loss of generality, we assume that $D \leq D'$, though some measures require that $D = D'$. In such cases, preprocessing techniques can be applied (see Appendix B). We note that this definition is limited to comparisons of *pairs* of representations, which is the standard setting in literature. In practice, one may also be interested in measuring similarity of *groups* of representations. The most direct way to obtain such measures of similarity for groups of representations from the standard pairwise measures is to aggregate the pairwise similarity scores, e.g., by averaging the similarities of all pairs of representations.

Typical issues when measuring similarity of representations are that the measures have to identify when a pair of representations is equivalent and that some measures may require preprocessing of the representations. In the following sections, we discuss these issues and related concepts in more detail.

Equivalence of Representations. Even if two representation matrices $R, R' \in \mathbb{R}^{N \times D}$ are not identical on an element-per-element basis, one may still consider them to be equivalent, i.e., perfectly similar. An intuitive example for such a case would be when representations only differ in their sign, i.e., $R = -R'$, or when representations can be rotated onto another. Such notions of equivalence can be formalized in terms of bijective mappings (transformations) $\varphi : \mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times D}$ that yield $\varphi(R) = R'$. What kind of transformations constitute equivalence between representations may vary depending on the context at hand. For instance, equivalence up to rotation does not make sense if some feature dimensions are already aligned with fixed axes, as is the case in interpretable word embeddings where axes may represent scales between polar opposites like “bright” and “dark” [94]. Thus, we define equivalence of representations in terms of groups of transformations $\mathcal{T} := \mathcal{T}(N, D)$ and call two representations R, R' equivalent with respect to a group \mathcal{T} , written as $R \sim_{\mathcal{T}} R'$, if there is a $\varphi \in \mathcal{T}$ such that $\varphi(R) = R'$.

In practice, it is crucial to determine under which groups of transformations representations should be considered equivalent, as equivalent representations should be indistinguishable for the chosen similarity measure. Conversely, representations that are not equivalent have to be

distinguishable for a similarity measure. In formal terms, this means that a measure has to be *invariant* to exactly those groups of transformations that the underlying representations are equivalent under. We call a representational similarity measure m invariant to a group of transformations \mathcal{T} if, for all $\mathbf{R} \in \mathbb{R}^{N \times D}$, $\mathbf{R}' \in \mathbb{R}^{N \times D'}$ and all $\varphi \in \mathcal{T}(N, D)$, $\varphi' \in \mathcal{T}(N, D')$, it holds that $m(\mathbf{R}, \mathbf{R}') = m(\varphi(\mathbf{R}), \varphi'(\mathbf{R}'))$. Thus, if a measure m is invariant to \mathcal{T} , then it directly follows that $m(\mathbf{R}, \mathbf{R}) = m(\mathbf{R}, \mathbf{R}')$ if $\mathbf{R} \sim_{\mathcal{T}} \mathbf{R}'$. This implies that a measure can only distinguish representations that are not equivalent under the groups of transformations it is invariant to. Using this notion of invariance, and assuming that representations have identical dimensionality, we can also analyze whether measures satisfy the criteria of a distance metric—in the context of representational similarity, these criteria are typically relaxed to only require $m(\mathbf{R}, \mathbf{R}') = 0$ if and only if $\mathbf{R} \sim_{\mathcal{T}} \mathbf{R}'$ for a group of transformations \mathcal{T} that m is invariant to [155, Apx. A.2].

In the literature [73, 81, 114, 155], there are six main groups of transformations under which representations are considered equivalent and that representational similarity measures are often designed to be invariant to the following:

- **Permutations (PT)**. A similarity measure m is invariant to permutations if swapping columns of the representation matrices \mathbf{R} , that is, reordering neurons, does not affect the resulting similarity score. Letting S_D denote the set of all permutations on $\{1, \dots, D\}$, and for $\pi \in S_D$, $\mathbf{P}_{\pi} = (p_{i,j}) \in \mathbb{R}^{D \times D}$ denote the permutation matrix where $p_{i,j} = 1$ if $\pi(i) = j$ and $p_{i,j} = 0$ otherwise, the group of all permutation transformations is given by

$$\mathcal{T}_{PT} = \{\mathbf{R} \mapsto \mathbf{R}\mathbf{P}_{\pi} : \pi \in S_D\}. \quad (3)$$

Permutations neither affect Euclidean distances nor angles between instance representations.

- **Orthogonal Transformations (OT)**. As noted in an earlier example, one might intuitively consider two representations equivalent if they can be rotated onto each other. Next to rotations, the group of orthogonal transformations also includes permutations and reflections. Letting $O(D) := \{\mathbf{Q} \in \mathbb{R}^{D \times D}, \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_D\}$ denote the orthogonal group, the set of these transformations is given by

$$\mathcal{T}_{OT} = \{\mathbf{R} \mapsto \mathbf{R}\mathbf{Q} : \mathbf{Q} \in O(D)\}. \quad (4)$$

These transformations preserve both Euclidean distances and angles between instance representations.

- **Isotropic Scaling (IS)**. Scaling all elements of a representation \mathbf{R} identically (isotropic scaling) does not change the angles between instance representations \mathbf{R}_i . The set of all isotropic scaling transformations is defined as

$$\mathcal{T}_{IS} = \{\mathbf{R} \mapsto a \cdot \mathbf{R} : a \in \mathbb{R}_+\}. \quad (5)$$

Isotropic scaling of representations will also rescale the Euclidean distance between instance representations by the same scaling factor a .

- **Invertible Linear Transformations (ILT)**. The group of invertible linear transformations, which is defined as

$$\mathcal{T}_{ILT} = \{\mathbf{R} \mapsto \mathbf{R}\mathbf{A} : \mathbf{A} \in GL(D, \mathbb{R})\}, \quad (6)$$

with $GL(D, \mathbb{R})$ denoting the general linear group of all invertible matrices $\mathbf{A} \in \mathbb{R}^{D \times D}$, forms a broader group of transformations. It includes both orthogonal transformations and rescalings. Both angles and Euclidean distances between instance representations are generally not preserved.

- **Translations (TR)**. If the angles between instance representations \mathbf{R}_i are not of concern, then one might argue that two representations are equivalent if they can be mapped onto

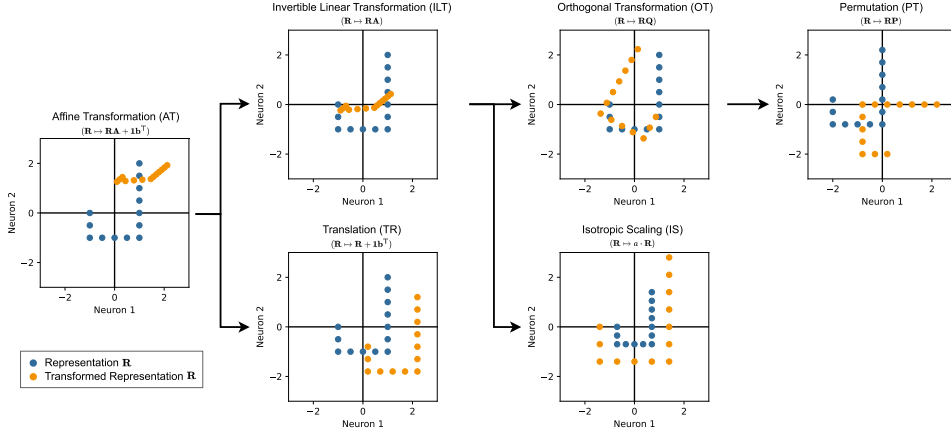


Fig. 2. Illustration of representations considered equivalent under different invariances. The invariances form a hierarchy: Arrows describe implication, with the left invariance being more general. For AT and ILT, the same linear transformation is applied. AT further translates representations by the same vector that is used in TR. In OT, the representations are rotated (120°) and reflected over the 15° axis. In PT, axes are swapped. IS applies a scaling factor of 2. See Appendix G for exact parameter values.

each other by adding a constant vector. In that regard, a measure m is invariant to translations if is invariant to the set of all mappings

$$\mathcal{T}_{\text{TR}} = \{R \mapsto R + \mathbf{1}_N \mathbf{b}^T : \mathbf{b} \in \mathbb{R}^D\}, \quad (7)$$

where $\mathbf{1}_N$ is a vector of N ones. Translations preserve Euclidean distances between instance representations.

- **Affine Transformations (AT).** The most general group of transformations that is typically considered for representations is given by the set of affine transformations

$$\mathcal{T}_{\text{AT}} = \{R \mapsto RA + \mathbf{1}_N \mathbf{b}^T : A \in \text{GL}(D, \mathbb{R}), \mathbf{b} \in \mathbb{R}^D\}. \quad (8)$$

This group of transformations in particular also includes rescaling, translations, orthogonal transformations, and invertible linear transformations. Therefore, affine transformations in general do neither preserve angles nor Euclidean distances between instance representations.

We depict the hierarchy of these groups in Figure 2. Table 1 also shows the invariances of all representational similarity measures covered in this survey with respect to these groups. This list of groups of transformations is, however, not exhaustive, and both for practical and theoretical reasons, various other groups may be considered [47]. In practice, neurons are typically assumed to be indexed arbitrarily, so most representational similarity measures are invariant to permutations [65, 73, 81]. Raghu et al. [114] further argued for invariance to invertible linear transformations, as any such transformation could be reverted by a directly following linear layer without altering overall network behavior. However, Kornblith et al. [73] criticized that such invariance leads to unintuitive similarity behavior when $D > N$, such as all representations with full rank being equivalent, and that training of neural networks is not invariant to linear transformations. They argued that orthogonal transformations capture practical differences in representations better.

Preprocessing of Representations. Many representational similarity measures assume certain properties of the representations R, R' that, in practice, are not always given. For instance, it

Table 1. Overview of Representational Similarity Measures

| Type | Measure | Invariances | | | | | | Preprocessing | $D \neq D'$ | Metric | Similarity \uparrow |
|------------------------------------|--|----------------|----------------|----------------|----------------|----------------|----------------|---------------|-------------|----------------|-----------------------|
| | | PT | OT | IS | ILT | TR | AT | | | | |
| Canonical Correlation Analysis | Mean Canonical Correlation [114] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | CC | ✓ | ✗ | ✓ |
| | Mean Squared Canonical Correlation [73, 159] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | CC | ✓ | ✗ | ✓ |
| | Singular Vector Canonical Correlation Analysis (SVCCA) [114] | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | CC | ✓ | ✗ | ✓ |
| | Projection-Weighted Canonical Correlation Analysis (PWCCA) [101] | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ | CC | ✓ | ✗ | ✓ |
| Alignment | Orthogonal Procrustes [34, 155] | ✓ | ✓ [*] | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| | Angular Shape Metric [155] | ✓ | ✓ [*] | ✓ | ✗ | ✗ | ✗ | MN | ✗ | ✓ | ✗ |
| | Partial Whitening Shape Metric [155] | ✓ | ✓ | ✓ [†] | ✓ [†] | ✓ | ✓ [†] | ✗ | ✗ | ✓ | ✗ |
| | Soft Matching Distance [65] | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ | CC, MN | ✓ | ✓ | ✗ |
| | Linear Regression [73, 81] | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | CC | ✓ | ✗ | ✗ |
| | Aligned Cosine Similarity [54] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| | Correlation Match [81] | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | CC | ✗ | ✗ | ✓ |
| | Maximum Matching Similarity [152] | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| | ContraSim [116] | ✓ [†] | ✓ [†] | ✓ [†] | ✓ [†] | ✓ [†] | ✓ [†] | ✗ | ✓ | ✗ | ✓ |
| Representational Similarity Matrix | Norm of Representational Similarity Matrix Difference [125, 162] | ✓ [‡] | ✓ [‡] | ✗ [‡] | ✗ | ✗ [‡] | ✗ | ✗ | ✓ | ✓ [‡] | ✗ |
| | Representational Similarity Analysis (RSA) [74] | ✓ [‡] | ✗ [‡] | ✓ [‡] | ✗ | ✓ [‡] | ✗ | ✗ | ✓ | ✗ | ✓ [‡] |
| | Centered Kernel Alignment (CKA) [73] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | CC | ✓ | ✗ | ✓ |
| | Distance Correlation (dCor) [139] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Normalized Bures Similarity (NBS) [141] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Eigenspace Overlap Score (EOS) [95] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Unified Linear Probing (GULP) [17] | ✓ | ✓ | ✓ | ✗ [‡] | ✓ | ✗ [‡] | CC, RN | ✓ | ✓ | ✗ |
| | Riemmanian Distance [125] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ |
| Neighbors | Relational Knowledge Loss [112] | ✓ | ✓ | ✓ [‡] | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| | k -NN Jaccard Similarity [52, 61, 123, 149] | ✓ | ✓ | ✓ | ✗ | ✗ [‡] | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Second-Order Cosine Similarity [53] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Rank Similarity [149] | ✓ | ✓ | ✓ | ✗ | ✗ [‡] | ✗ | ✗ | ✓ | ✗ | ✓ |
| Topology | Joint Rank and Jaccard Similarity [149] | ✓ | ✓ | ✓ | ✗ | ✗ [‡] | ✗ | ✗ | ✓ | ✗ | ✓ |
| | Geometry Score [66] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| | Multi-Scale Intrinsic Distance (IMD) [144] | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Statistic | Representation Topology Divergence (RTD) [8] | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| | Intrinsic Dimension [21] | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | § |
| | Magnitude [149] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | § |
| | Concentricity [149] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | § |
| | Uniformity [153] | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | § |
| | Tolerance [150] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | RN | ✓ | ✗ | § |
| | Instance-Graph Modularity [87, 122] | ✓ | ✓ | ✓ [†] | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | § |
| | Neuron-Graph Modularity [77] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | § |

*: Subgroups possible. †: Varies based on hyperparameters. ‡: Similarity function dependent. §: Depends on comparison.

The invariances are permutation (PT), orthogonal transformation (OT), isotropic scaling (IS), invertible linear transformation (ILT), translation (TR), and affine transformation (AT). We report invariances based on default hyperparameters and preprocessing as proposed by the authors. These may vary if different parameters or similarity functions are applied. Three kinds of preprocessing are commonly used: centering columns (CC), normalizing the matrix norm (MN), or normalizing row norms (RN). The column $D \neq D'$ indicates whether a measure requires the compared representations to have identical dimensionality. *Metric* indicates whether a similarity measure satisfies the criteria of a distance metric when representations have equal dimensionality. *Similarity* \uparrow indicates whether increasing scores imply increasing similarity of models.

is often assumed that representations are mean centered in the columns [73, 101, 155] or that they have the same dimensionality. In these cases, the representations need to be preprocessed. There are three kinds of preprocessing that may have to be applied, namely normalization, adjusting dimensionality, and flattening of representations. We discuss these problems in Appendix B.

2.2 Functional Similarity Measures

Functional similarity measures compare neural networks by measuring similarity of their output behavior [30]. Given a set of inputs X and a neural network f that is trained for a classification task on C classes, we let

$$\mathbf{O} := f(X) \in \mathbb{R}^{N \times C} \quad (9)$$

denote the matrix of its outputs. Each row $\mathbf{O}_i = f(X_i) \in \mathbb{R}^C$ corresponds to the output for input X_i . In the context of this survey, we assume that this vector-based output corresponds to *soft predictions*, where each element $\mathbf{O}_{i,c}$ denotes the probabilities or decision scores of class c for

input X_i . From these soft predictions, we can compute the *hard predictions* for a given multiclass classification task via $\hat{c} = \arg \max_c \mathbf{O}_{i,c}$, where \hat{c} denotes the predicted class for input X_i .

Then, similarly to representational similarity measures, functional similarity measures are defined as mappings $m : \mathbb{R}^{N \times C} \times \mathbb{R}^{N \times C} \rightarrow \mathbb{R}$ that assign a similarity score $m(\mathbf{O}, \mathbf{O}')$ to a pair of outputs \mathbf{O}, \mathbf{O}' , which are derived from the same inputs \mathbf{X} . For the compared outputs $\mathbf{O}, \mathbf{O}' \in \mathbb{R}^{N \times C}$, it is assumed that they are aligned in the sense that the columns $\mathbf{O}_{-,c}, \mathbf{O}'_{-,c}$ correspond to probability/decision scores of the same class c .

Due to this alignment and the fixed semantics of outputs, analyzing functional similarity generally does not require consideration of preprocessing or invariances. For the same reason, representational similarity measures are unsuitable for comparison of outputs—the previous assumptions do not hold for representations. Thus, for instance, all permutation invariant measures would consider two outputs that assign 100% probability to different classes equivalent.

Moreover, many functional similarity measures require only black-box access to a model, relying solely on knowledge about inputs and outputs. However, functional similarity measures may include additional information aside from the raw outputs \mathbf{O}_i . For instance, a set of ground-truth labels $\mathbf{y} \in \mathbb{R}^N$ is often given, which is typically used by a quality function q that quantifies how well the output matches the ground truth. Another kind of additional information are task-based gradients, which, however, require white-box access to the model. Finally, in the context of functional similarity, it is more common that measures compare multiple models at once without relying on pairwise comparisons.

2.3 Relationship between Representational and Functional Similarity

The notions of representational and functional similarity complement each other (see Figure 1), and applying both representational and functional similarity measures allows for a more holistic view of neural network similarity (e.g., References [52, 69, 137]). To properly interpret potentially conflicting similarity scores stemming from these two perspectives, it is crucial to understand their relationship.

When functional similarity measures indicate dissimilarity, representations must be dissimilar at some layer, assuming that differences in the final classification layer cannot fully explain the functional difference. The opposite is not true: Two functionally similar models may use dissimilar representations. Even more, if a functional similarity measure indicates high similarity on a given input set, then this does not imply that the compared models are functionally similar in general: High similarity may be due to easy-to-classify inputs, and out-of-distribution inputs, which tend to amplify functional differences, could yield lower similarity in the corresponding outputs. Similarly, a representational measure indicating high similarity might not generally indicate high functional or representational similarity between models either, as the invariance of a measure might not fit to the given representations.

In conclusion, one generally cannot expect functional and representational measures to correlate and their scores require contextualization. Only if there is significant functional dissimilarity between two models should there also be a representational measure indicating significant dissimilarity. Since functional outputs and their similarity measures have a clear and intuitive semantic, this relation can also be used to validate representational similarity measures [34].

3 Representational Similarity Measures

We now review existing representational similarity measures, categorized by their underlying approach to measuring similarity. The categories are illustrated in Figure 3. An overview of all reviewed representational similarity measures can be found in Table 1.

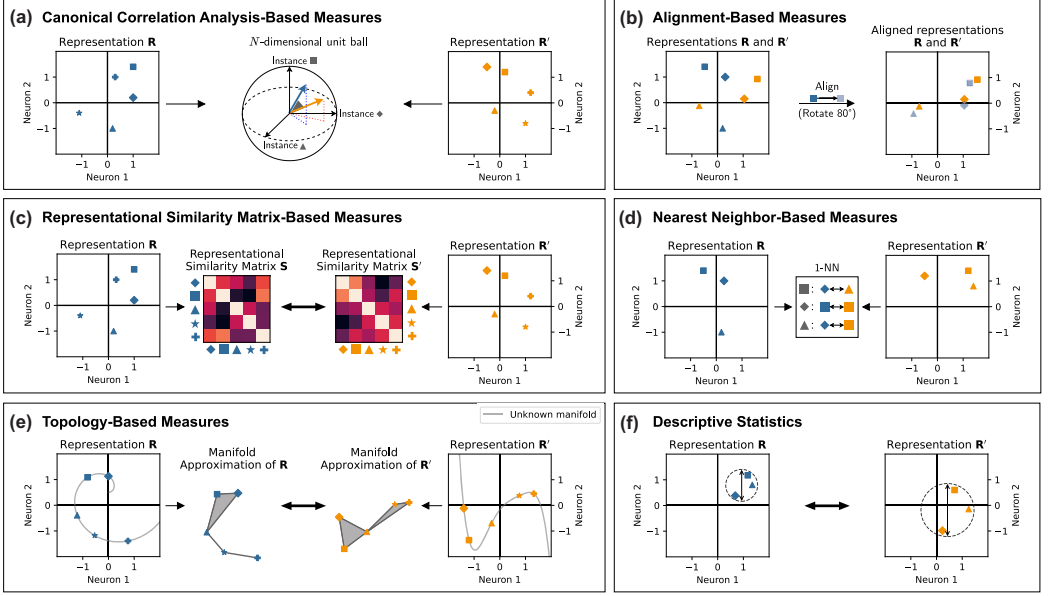


Fig. 3. Types of representational similarity measures, illustrated with two-dimensional representations. (a) Representations of N instances are projected onto the N -dimensional unit ball, and similarity is then quantified based on their angle (their correlation). The illustration of the unit ball is not to scale, and only the first three dimensions are shown. (b) Representations are aligned with each other, and similarity is computed after alignment. (c) Similarity is based on comparing matrices of pairwise similarities within representations. (d) Representations are compared based on similarity of their k nearest neighbors, here $k = 1$. (e) Manifolds of the representations are approximated and compared. (f) Statistics are computed individually for each representation (here: spread of instance representations) and then compared.

3.1 Canonical Correlation Analysis-based Measures

Canonical Correlation Analysis (CCA) [60] is a classical method to compare two sets of values of random variables. CCA finds weights $\mathbf{w}_R \in \mathbb{R}^D$, $\mathbf{w}_{R'} \in \mathbb{R}^{D'}$ for the columns in the representations, such that the linear combinations $R\mathbf{w}_R$ and $R'\mathbf{w}_{R'} \in \mathbb{R}^N$ have maximal correlation. Geometrically, the vectors $\mathbf{w}_R, \mathbf{w}_{R'}$ are projected to the unit ball in \mathbb{R}^N via their representation matrices, such that their angle is minimal. Assuming mean-centered representations, the first *canonical correlation* ρ is defined as

$$\rho := \rho(R, R') := \max_{\mathbf{w}_R, \mathbf{w}_{R'}} \frac{\langle R\mathbf{w}_R, R'\mathbf{w}_{R'} \rangle}{\|R\mathbf{w}_R\| \cdot \|R'\mathbf{w}_{R'}\|}. \quad (10)$$

One can find additional canonical correlations ρ_i , that are uncorrelated and thus orthogonally projected to the previous ones. This yields a system of D canonical correlations ρ_i defined as

$$\rho_i := \max_{\mathbf{w}_R^{(i)}, \mathbf{w}_{R'}^{(i)}} \frac{\langle R\mathbf{w}_R^{(i)}, R'\mathbf{w}_{R'}^{(i)} \rangle}{\|R\mathbf{w}_R^{(i)}\| \cdot \|R'\mathbf{w}_{R'}^{(i)}\|} \quad \text{s.t. } R\mathbf{w}_R^{(j)} \perp R\mathbf{w}_R^{(i)}, R'\mathbf{w}_{R'}^{(j)} \perp R'\mathbf{w}_{R'}^{(i)} \quad \forall j < i, \quad (11)$$

where \perp means orthogonality. If the representations are (nearly) collinear, then regularized *Ridge CCA* [147] can be used.

A single similarity score $m(R, R')$ is then computed by aggregating the canonical correlations ρ_i . Standard aggregation choices used to quantify neural network similarity are the mean canonical correlation m_{CCA} [57, 73, 114] and the mean squared canonical correlation m_{CCA^2} [57, 73], also

called *Yanai's generalized coefficient of determination* [159],

$$m_{\text{CCA}}(\mathbf{R}, \mathbf{R}') = \frac{1}{D} \sum_{i=1}^D \rho_i, \quad m_{\text{CCA}^2}(\mathbf{R}, \mathbf{R}') = \frac{1}{D} \sum_{i=1}^D \rho_i^2. \quad (12)$$

CCA is invariant to affine transformations [101]. If the representations \mathbf{R}, \mathbf{R}' are equivalent, then it holds that $\rho_i = 1$ for all $i \in \{1, \dots, D\}$ and thus $m_{\text{CCA}}(\mathbf{R}, \mathbf{R}') = 1$ and $m_{\text{CCA}^2}(\mathbf{R}, \mathbf{R}') = 1$.

Other prominent aggregation schemes, though not applied for representational similarity, include the sum of the squared canonical correlations (also known as Pillai's trace [113]), Wilk's lambda statistic [154], and the Lawley–Hotelling trace [59, 78]. Several more aggregation methods can be applied, and there are numerous variants of CCA measures, including non-linear and multiview ones—overviews on such variants are provided in the recent survey by Yang et al. [160] or the tutorial by Uurtio et al. [146]. In this work, however, we only consider those CCA-based measures that have been used to measure representational similarity of neural networks.

Singular Value CCA. Raghu et al. [114] argued that representations are noisy and that this noise should be removed before conducting CCA on the representations \mathbf{R}, \mathbf{R}' . Thus, they proposed the *Singular Value CCA* (SVCCA) approach, in which denoised representations are obtained by performing *principal component analysis* (PCA) on the representations. The number k of principal components that are kept is selected such that a fixed relative amount t of the variance in the data, usually 99%, is explained. Afterward, they use standard CCA on the denoised representations. Thus, letting $\tilde{\mathbf{R}}, \tilde{\mathbf{R}'}$ denote the denoised representations, the average canonical correlation is used as the final similarity measure:

$$m_{\text{SVCCA}}(\mathbf{R}, \mathbf{R}') = m_{\text{CCA}}(\tilde{\mathbf{R}}, \tilde{\mathbf{R}'}). \quad (13)$$

Practically, the representations are also mean-centered before the PCA denoising. Unlike CCA, SVCCA is only invariant to orthogonal transformations, isotropic scaling and translation. SVCCA is bounded in the interval $[0, 1]$, with a score of one indicating perfectly similar representations.

To compute SVCCA efficiently for *convolutional neural networks* (CNNs) with many features, Raghu et al. [114] applied a Discrete Fourier Transform on each channel, yielding block-diagonal matrices for CCA computation, which eliminates unneeded operations.

Projection Weighted CCA. Morcos et al. [101] proposed *Projection Weighted CCA* (PWCCA) as an alternative to SVCCA. They argued that a representational similarity measure should weigh the individual canonical correlations ρ_i by their importance, i.e., the similarity of the *canonical variables* $\mathbf{R}\mathbf{w}_R^{(i)}$ with the raw representation \mathbf{R} .

For that purpose, given mean-centered representations, they defined a weighting coefficient $\tilde{\alpha}_i = \sum_{j=1}^D |\langle \mathbf{R}\mathbf{w}_R^{(i)}, \mathbf{R}_{-,j} \rangle|$ for every canonical correlation ρ_i that models its importance. These coefficients are then normalized to weights $\alpha_i = \tilde{\alpha}_i / \sum_j \tilde{\alpha}_j$, yielding the final representational similarity measure

$$m_{\text{PWCCA}}(\mathbf{R}, \mathbf{R}') = \sum_{i=1}^D \alpha_i \rho_i. \quad (14)$$

This measure is asymmetric, since the weights α_i are only computed based on \mathbf{R} . Further, it is invariant to isotropic scaling and translation. PWCCA is bounded in the interval $[0, 1]$, with a value of one indicating equivalent representations.

3.2 Alignment-based Measures

The next group of measures stipulates that a pair of representations \mathbf{R}, \mathbf{R}' can be compared directly once the corresponding representation spaces have been aligned to each other. Alignment is usually realized by finding an optimal transformation $\varphi \in \mathcal{T}$ that minimizes a difference of the form $\|\varphi(\mathbf{R}) - \mathbf{R}'\|$. The exact group of transformations \mathcal{T} used for alignment also directly determines and usually corresponds to the group of transformations that the corresponding measure will be invariant to. Such direct alignment is only possible if the number of neurons in both representations

are equal. Thus, we assume throughout the next section that $D = D'$, unless otherwise mentioned. We now discuss existing measures from this category.

Orthogonal Procrustes. The orthogonal Procrustes problem is a classical problem of finding the best orthogonal transformation to align two matrices in terms of minimizing the Frobenius norm (Equation (A.2)) of the difference. Solving the problem leads to the similarity measure

$$m_{\text{Ortho-Proc}}(\mathbf{R}, \mathbf{R}') = \min_{\mathbf{Q} \in \text{O}(D)} \|\mathbf{R}\mathbf{Q} - \mathbf{R}'\|_F = (\|\mathbf{R}\|_F^2 + \|\mathbf{R}'\|_F^2 - 2\|\mathbf{R}^\top \mathbf{R}'\|_*)^{\frac{1}{2}}, \quad (15)$$

where $\|\cdot\|_*$ denotes the nuclear norm (Equation (A.3)) of a matrix [124]. The second formulation can also be used if $D \neq D'$ [65]. Ding et al. [34] used the square of $m_{\text{Ortho-Proc}}$ as a similarity score. By design, this measure is invariant to orthogonal transformations, and Williams et al. [155] showed that this measure satisfies the properties of a distance metric. This also holds when one optimizes Equation (15) over any subgroup $G(D) \subset \text{O}(D)$. Notably, considering the subgroup of permutation matrices yields the *Permutation Procrustes* measure [155], also known as *one-to-one matching distance* [65].

A similar optimization was proposed by Godfrey et al. [47] in their G_{ReLU} -Procrustes measure, which is designed to be invariant to G_{ReLU} transformations, a special set of linear transformations (see Appendix A.4). Williams et al. [155] further proposed a variant that is invariant to spatial shifts in convolutional layers.

Generalized Shape Metrics. Williams et al. [155] applied theory of statistical shape analysis on the problem of measuring representational similarity. In that context, they also defined novel similarity measures. For representations with unit Frobenius norm (Equation (A.2)) and any subgroup $G(D) \subseteq \text{O}(D)$, they introduced the *Angular Shape Metric*,

$$m_\theta(\mathbf{R}, \mathbf{R}') = \min_{\mathbf{Q} \in G(D)} \arccos \langle \mathbf{R}\mathbf{Q}, \mathbf{R}' \rangle_F, \quad (16)$$

which is invariant to transformations from $G(D)$. To obtain a more general measure that is not restricted to representations preprocessed to unit norm, they apply the partial whitening function $\phi_\alpha(\mathbf{R}) = \mathbf{H}_N \mathbf{R} (\alpha \mathbf{I}_D + (1 - \alpha)(\mathbf{R}^\top \mathbf{H}_N \mathbf{R})^{-1/2})$, where $\alpha \in [0, 1]$ and $\mathbf{H}_N = \mathbf{I}_N - \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^\top$ denotes a centering matrix. This yields the *Partial Whitening Shape Metric*,

$$m_{\theta, \alpha}(\mathbf{R}, \mathbf{R}') = \min_{\mathbf{Q} \in \text{O}(D)} \arccos \frac{\langle \phi_\alpha(\mathbf{R})\mathbf{Q}, \phi_\alpha(\mathbf{R}') \rangle_F}{\|\phi_\alpha(\mathbf{R})\|_F \|\phi_\alpha(\mathbf{R}')\|_F}. \quad (17)$$

For all $\alpha > 0$, this metric is invariant to orthogonal transformations and translations. For $\alpha = 1$, it is further invariant to isotropic scaling, for $\alpha = 0$ it is even invariant to affine transformations. Williams et al. [155] showed that this metric is also related to (regularized) canonical correlations. Both shape metrics are bounded in the interval $[0, \pi]$ and satisfy the properties of a distance metric.

Duong et al. [37] generalized the metrics from Williams et al. [155] to stochastic neural networks such as variational autoencoders [67], which map to distributions of representations instead of deterministic representations. Ostrow et al. [109] further extended this metric to measure similarity of dynamical systems, such as RNNs.

Soft Matching Distance. Khosla and Williams [65] generalized the Permutation Procrustes measure to settings in which the number of neurons in the representations \mathbf{R}, \mathbf{R}' differ, i.e., $D \neq D'$. This was done by interpreting the problem of matching neurons as a transportation problem with possible solutions in the transportation polytope $\text{TP}(D, D')$ [32]. Thus, assuming the representations are centered and scaled to unit norm, they defined the *soft matching distance* as

$$m_{\text{SoftMatch}}(\mathbf{R}, \mathbf{R}') = \sqrt{\min_{\mathbf{P} \in \text{TP}(D, D')} \sum_{ij} \mathbf{P}_{ij} \|\mathbf{R}_{-,i} - \mathbf{R}'_{-,j}\|_2^2}. \quad (18)$$

This measure is a special case of the 2-Wasserstein distance and thus a metric [65]. Further, it is invariant to permutations, translations, and scaling. Khosla and Williams [65] also proposed a variant that is related to Correlation Match (Equation (21)).

Linear Regression. An approach similar to Procrustes, but not restricted to orthogonal transformations, is based on predicting one representation from the other with a linear transformation [73, 81]. Then, the R-squared score of the optimal fit can be used to measure similarity [73]. Assuming mean-centered representations, this yields the measure

$$m_{R^2}(\mathbf{R}, \mathbf{R}') = 1 - \frac{\min_{\mathbf{W} \in \mathbb{R}^{D \times D}} \|\mathbf{R}' - \mathbf{R}\mathbf{W}\|_F^2}{\|\mathbf{R}'\|_F^2} = \frac{\|(\mathbf{R}'(\mathbf{R}'^T \mathbf{R}')^{-1/2})^T \mathbf{R}\|_F^2}{\|\mathbf{R}\|_F^2}. \quad (19)$$

This asymmetric measure is invariant to orthogonal transformation and isotropic scaling. A value of one indicates maximal similarity, lower values indicate lower similarity. This measure has no lower bound.

Li et al. [81] added a L1 penalty to the optimization to encourage a sparse mapping between neurons. Bau et al. [10] matched the full representation of one model to a single neuron of another by linear regression.

Aligned Cosine Similarity. This measure was used to quantify similarity of instance representations, such as embeddings of individual words over time [54]. Its idea is to first align the representations by the orthogonal Procrustes transformation and then to use cosine similarity (Equation (A.5)) to measure similarity between the aligned representations. Letting \mathbf{Q}^* denote the solution to the Procrustes problem (Equation (15)), the similarity of two instance representations is given by $\cos\text{-sim}((\mathbf{R}\mathbf{Q}^*)_i, \mathbf{R}'_i)$. Overall similarity can then be analyzed by comparing the overall distribution of similarity scores or aggregating them by, for instance, taking their mean value [123]. The latter option yields a similarity measure

$$m_{\text{Aligned-Cosim}}(\mathbf{R}, \mathbf{R}') = \frac{1}{N} \sum_{i=1}^N \cos\text{-sim}((\mathbf{R}\mathbf{Q}^*)_i, \mathbf{R}'_i), \quad (20)$$

which is bounded in the interval $[-1, 1]$, with $m_{\text{Aligned-Cosim}}(\mathbf{R}, \mathbf{R}') = 1$ indicating perfect similarity. It is invariant to orthogonal transformations and isotropic scaling.

Correlation Match. Li et al. [81] measured representational similarity by creating a correlation matrix between the neuron activations of two representations that are assumed to be mean-centered. They then matched each neuron $\mathbf{R}_{-,j}$ to the neuron $\mathbf{R}'_{-,k}$ that it correlated the strongest with. Wu et al. [156] applied strict one-to-one matching, Li et al. [81] further used a relaxed version, in which one neuron can correspond to multiple other ones. Letting \mathbf{M} denote the matrix that matches the neurons, which is a permutation matrix in strict one-to-one matching, the average correlation between the matched neurons is given by

$$m_{\text{Corr-Match}}(\mathbf{R}, \mathbf{R}') = \frac{1}{D} \sum_{j=1}^D \frac{\langle \mathbf{R}_{-,j}, (\mathbf{R}'\mathbf{M})_{-,j} \rangle}{\|\mathbf{R}_{-,j}\|_2 \|(\mathbf{R}'\mathbf{M})_{-,j}\|_2}. \quad (21)$$

This measure is invariant to permutations, isotropic scaling, and translations. A value of one indicates equivalent representations, a value of zero uncorrelated ones.

Maximum Matching Similarity. In contrast to the previous measures, *Maximum Matching Similarity* [152] aligns representations only implicitly and can compare representations of different dimension by testing whether neuron activations of one representation, i.e., columns of the representation matrix, (approximately) lie in a subspace spanned from neuron activations of the other representation. Every neuron, of which the activation vector can be approximated by such a subspace, is then considered part of a match between the representations. Following this intuition, the main idea of the measure proposed by Wang et al. [152] is to find the maximal set of neurons in each representation that can be matched with the other subspace. Formally, for an index subset

$\mathcal{J} \subseteq \{1, \dots, D\}$, let $\mathbf{R}_{-, \mathcal{J}} = \{\mathbf{R}_{-, j}, j \in \mathcal{J}\}$ denote the set of corresponding neuron activation vectors. Then a pair $(\mathcal{J}, \mathcal{J}')$ forms an ε -approximate match, $\varepsilon \in (0, 1]$, on the representations \mathbf{R}, \mathbf{R}' if for all $j \in \mathcal{J}, j' \in \mathcal{J}'$ it holds that

$$\min_{\mathbf{r} \in \text{span}(\mathbf{R}_{-, \mathcal{J}})} \|\mathbf{R}'_{-, j'} - \mathbf{r}\| \leq \varepsilon \cdot \|\mathbf{R}'_{-, j'}\| \quad \text{and} \quad \min_{\mathbf{r}' \in \text{span}(\mathbf{R}'_{-, \mathcal{J}'})} \|\mathbf{R}_{-, j} - \mathbf{r}'\| \leq \varepsilon \cdot \|\mathbf{R}_{-, j}\|. \quad (22)$$

A pair $(\mathcal{J}_{\max}, \mathcal{J}'_{\max})$ is considered a maximum match, if for all ε -matches $(\mathcal{J}, \mathcal{J}')$ it holds that $\mathcal{J} \subseteq \mathcal{J}_{\max}$ and $\mathcal{J}' \subseteq \mathcal{J}'_{\max}$. Wang et al. [152] showed that the maximum match is unique and provided algorithms to determine it. Based on the maximum match, the *maximum matching similarity* is defined as

$$m_{\text{maximum-match}}^{\varepsilon}(\mathbf{R}, \mathbf{R}') = \frac{|\mathcal{J}_{\max}| + |\mathcal{J}'_{\max}|}{D + D'}. \quad (23)$$

This measure is invariant to invertible linear transformation, since such transformations do not alter the subspaces. It is bounded in the interval $[0, 1]$, with a similarity score of 1 indicating maximum similarity.

ContraSim. Inspired by ideas from contrastive learning, Rahamim and Belinkov [116] proposed a measure that implicitly aligns representations by applying a neural encoder to map them into a joint embedding space. The encoder was trained using explicitly selected pairs of instances as positive and negative samples, for which the resulting embeddings should and should not be similar, respectively. In the joint embedding space, similarity is then modeled by the angle between representations, and thus, letting enc denote the trained encoder network, *ContraSim* is defined as

$$m_{\text{ContraSim}}(\mathbf{R}, \mathbf{R}') = \frac{1}{N} \sum_{i=1}^N \cos\text{-sim}(\text{enc}(\mathbf{R}_i), \text{enc}(\mathbf{R}'_i)). \quad (24)$$

If \mathbf{R}, \mathbf{R}' have different dimensionality, then two different encoders are trained together. The invariances of the measure are determined via the training examples for the encoder. The measure is bounded in the interval $[-1, 1]$, with a similarity score of 1 indicating maximum similarity.

3.3 Representational Similarity Matrix-based Measures

A common approach to avoid alignment issues in direct comparisons of representations is to use *representational similarity matrices* (RSMs). Intuitively, an RSM describes the similarity of the representation of each instance i to all other instances in a given representation \mathbf{R} . The RSMs of two representations \mathbf{R}, \mathbf{R}' can then be used to quantify representational similarity in terms of the difference between these RSMs. Formally, given an instancewise similarity function $s : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$, the RSM $S \in \mathbb{R}^{N \times N}$ of a representation \mathbf{R} can be defined in terms of its elements via

$$S_{i,j} := s(\mathbf{R}_i, \mathbf{R}_j). \quad (25)$$

Each row S_i then corresponds to the similarity between the representations of instance i and the representations of all other inputs, including itself. RSMs can be computed with a variety of similarity functions s such as cosine similarity [24] or kernel functions [73]—like before, we do not differentiate between the equivalent concepts of similarity and distance functions. Naturally, the choice of the underlying similarity function s impacts the kind of transformations that the representational similarity measures m will be invariant to: If the RSM is unchanged by a transformation, then the representational similarity will not change either. In Appendix A.3, we give an overview of commonly used similarity functions, along with the invariances they induce on the RSMs. After selecting a suitable similarity function s , two RSMs S, S' are compared. In the following, we review existing measures that use this approach.

Norm of RSM Difference. A direct approach to compare RSMs is to apply some matrix norm $\|\cdot\|$ to the difference between RSMs to obtain a measure

$$m_{\text{Norm}}(\mathbf{R}, \mathbf{R}') = \|S - S'\|. \quad (26)$$

which assigns a score of zero to equivalent representations, and higher scores to dissimilar representations. To compute the RSMs, Shahbazi et al. [125] and Yin and Shen [162] used the linear kernel. In that case, this measure is invariant to orthogonal transformations and satisfies the properties of a distance metric for representations of equal dimensionality.

Representational Similarity Analysis. Kriegeskorte et al. [74] proposed *Representational Similarity Analysis (RSA)* in neuroscience. RSA is a general framework that utilizes RSMs to compare sets of measurements, such as neural representations. In the first step of this framework, RSMs are computed with respect to an inner similarity function s_{in} . Since the RSMs are symmetric, their lower triangles can then be vectorized in a next step to vectors $\mathbf{v}(\mathbf{S}) \in \mathbb{R}^{N(N-1)/2}$. Finally, these vectors are compared by an outer similarity function s_{out} :

$$m_{\text{RSA}}(\mathbf{R}, \mathbf{R}') = s_{\text{out}}(\mathbf{v}(\mathbf{S}), \mathbf{v}(\mathbf{S}')). \quad (27)$$

This framework can be instantiated with various choices for the similarity functions s_{in} and s_{out} . This choice, however, affects the kind of transformations that RSA is invariant to, and further determines the range and interpretation of this measure. Kriegeskorte et al. [74] used Pearson correlation (Equation (A.8)) as inner similarity function s_{in} to compute the RSMs, and Spearman correlation as outer similarity function s_{out} , since these correlation measures induce invariance to scaling and translations. Kriegeskorte et al. [74] further suggested functions such as Euclidean or Mahalanobis distance.

Centered Kernel Alignment. Kornblith et al. [73] proposed *Centered Kernel Alignment (CKA)* [28, 29] to measure representational similarity. CKA uses kernel functions on mean-centered representations to compute the RSMs, which are then compared via the *Hilbert–Schmidt Independence Criterion (HSIC)* [50]. Given two RSMs \mathbf{S}, \mathbf{S}' , the HSIC can be computed via $\text{HSIC}(\mathbf{S}, \mathbf{S}') = \frac{1}{(N-1)^2} \text{tr}(\mathbf{S}\mathbf{H}_N\mathbf{S}'\mathbf{H}_N)$, where $\mathbf{H}_N = \mathbf{I}_N - \frac{1}{N}\mathbf{1}_N\mathbf{1}_N^\top$ denotes a centering matrix. Recent work [103] highlights the importance of using the debiased HSIC estimator of Song et al. [132], especially when $N < D$. Then, a normalization of the HSIC yields the CKA measure:

$$m_{\text{CKA}}(\mathbf{R}, \mathbf{R}') = \frac{\text{HSIC}(\mathbf{S}, \mathbf{S}')}{\sqrt{\text{HSIC}(\mathbf{S}, \mathbf{S})\text{HSIC}(\mathbf{S}', \mathbf{S}')}}. \quad (28)$$

CKA is bounded in the interval $[0, 1]$, with $m_{\text{CKA}}(\mathbf{R}, \mathbf{R}') = 1$ indicating equivalent representations. Kornblith et al. [73] computed the RSMs from the linear kernel and tested the *radial basis function (RBF)* kernel without reporting large differences in results. Saini et al. [122] used so-called affinity matrices, which result from sparse subspace clustering [38] of the representations, instead of RSMs. The standard linear version is invariant to orthogonal transformations and isotropic scaling.

CKA with linear kernel is equivalent to the RV coefficient, a statistical measure to compare data matrices [73, 119]. It can also be seen as a variant of PWCCA (Equation (14)) with an alternative weighting scheme, with the advantage that it does not require a matrix decomposition to be computed [73]. Further, Godfrey et al. [47] proposed the G_{ReLU} -CKA variant that is specific to models that use ReLU activations and invariant to G_{ReLU} transformations (Equation (A.10)).

Distance Correlation. *Distance Correlation (dCor)* [139] is a non-linear correlation measure that tests dependence of two vector-valued random variables X and Y with finite mean. In the context of our survey, we consider the instance representations as samples of such random variables. To determine the distance correlation of two representation matrices \mathbf{R}, \mathbf{R}' , one first computes the RSMs \mathbf{S}, \mathbf{S}' using Euclidean distance as similarity function s . Next, the RSMs are mean-centered in both rows and columns, which yields $\tilde{\mathbf{S}}, \tilde{\mathbf{S}'}$. Then the squared sample distance covariance of the RSMs \mathbf{S}, \mathbf{S}' can be computed via $\text{dCov}^2(\mathbf{S}, \mathbf{S}') = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \tilde{\mathbf{S}}_{i,j} \tilde{\mathbf{S}}'_{i,j}$. Finally, the squared distance

correlation is defined as

$$m_{\text{dCor}}^2(\mathbf{R}, \mathbf{R}') = \frac{\text{dCov}^2(\mathbf{S}, \mathbf{S}')}{\sqrt{\text{dCov}^2(\mathbf{S}, \mathbf{S}) \text{dCov}^2(\mathbf{S}', \mathbf{S}')}}. \quad (29)$$

A distance correlation of zero indicates statistical independence between the representations \mathbf{R} and \mathbf{R}' . Due to the usage of Euclidean distance as similarity function s , dCor is invariant to orthogonal transformations and translations.

Lin [83] considered a variant called Adaptive Geo-Topological Independence Criterion [84], which rescales the values in the RSMs with respect to an upper and lower threshold to eliminate noise.

Normalized Bures Similarity. This measure was inspired by the Bures distance, which has its roots in quantum information theory [20] and satisfies the properties of a distance metric on the space of positive semidefinite matrices [13]. As Tang et al. [141] used the linear kernel to compute the RSMs \mathbf{S}, \mathbf{S}' , these matrices are positive semidefinite. Hence, these matrices also have a unique square root. Therefore, they could define the *Normalized Bures Similarity (NBS)* as

$$m_{\text{NBS}}(\mathbf{R}, \mathbf{R}') = \frac{\text{tr}(\mathbf{S}^{1/2} \mathbf{S}' \mathbf{S}^{1/2})^{1/2}}{\sqrt{\text{tr}(\mathbf{S}) \text{tr}(\mathbf{S}')}}. \quad (30)$$

This measure is bounded in the interval $[0, 1]$, with $m_{\text{NBS}}(\mathbf{R}, \mathbf{R}') = 1$ indicating perfect similarity. Due to use of the linear kernel, it is invariant to orthogonal transformations and further invariant to isotropic scaling due to the normalization.

One can show that NBS is equivalent—up to an arc cosine—to the angular shape metric (Equation (16)), and that the unnormalized Bures distance is equal to the orthogonal Procrustes measure [55].

Eigenspace Overlap Score. May et al. [95] proposed the *Eigenspace Overlap Score (EOS)* as a criterion to select compressed word embeddings with best downstream performance. EOS compares RSMs by comparing the spaces spanned from their eigenvectors. Assuming full-rank representations \mathbf{R}, \mathbf{R}' , they compute the RSMs \mathbf{S}, \mathbf{S}' using the linear kernel (Equation (A.6)). Letting $\mathbf{U} \in \mathbb{R}^{N \times D}, \mathbf{U}' \in \mathbb{R}^{N \times D'}$ denote the matrices of eigenvectors that correspond to the non-zero eigenvalues of \mathbf{S}, \mathbf{S}' , respectively, the measure is defined as

$$m_{\text{EOS}}(\mathbf{R}, \mathbf{R}') = \frac{1}{\max(D, D')} \|\mathbf{U}^T \mathbf{U}'\|_F^2. \quad (31)$$

EOS indicates minimal similarity with a value of zero when the spans of \mathbf{U} and \mathbf{U}' are orthogonal, and maximal similarity with a value of one when the spans are identical. This measure is invariant to invertible linear transformations.

EOS is related to the expected difference in generalization error of two linear models that are each trained on one of the representations [95], similar to the following measure.

Unified Linear Probing (GULP). *GULP* quantifies similarity by measuring how differently linear regression models that use either the representation \mathbf{R} or the representation \mathbf{R}' [17] can generalize. This is done by considering all regression functions η on the original instances \mathbf{X} that are bounded so that $\|\eta\|_{L^2} \leq 1$, and trying to replicate these relations via ridge regression on the representations \mathbf{R} and \mathbf{R}' . The similarity measure is then defined as the supremum of the expected discrepancy in the predictions of ridge regression models trained to approximate η with \mathbf{R} or \mathbf{R}' as inputs, taken over all regression functions η .

Practically, Boix-Adsera et al. [17] proved that there is a closed form expression to estimate this value in terms of the covariance matrices of the representations, where it is assumed that the representations are mean-centered in the columns and that their rows have unit norm. Letting the RSMs $\mathbf{S} = \frac{1}{N} \mathbf{R}^T \mathbf{R}$ denote the matrix of covariance within a representation, $\mathbf{S}_{\mathbf{R}, \mathbf{R}'} = \frac{1}{N} \mathbf{R}^T \mathbf{R}'$ the cross-covariance matrix, and $\mathbf{S}^{-\lambda} = (\mathbf{S} + \lambda \mathbf{I}_D)^{-1}$ the inverse of a regularized covariance matrix, the

GULP measure can be computed as

$$m_{\text{GULP}}^{\lambda}(\mathbf{R}, \mathbf{R}') = \left(\text{tr}(\mathbf{S}^{-\lambda} \mathbf{S} \mathbf{S}^{-\lambda} \mathbf{S}) + \text{tr}(\mathbf{S}'^{-\lambda} \mathbf{S}' \mathbf{S}'^{-\lambda} \mathbf{S}') - 2 \text{tr}(\mathbf{S}^{-\lambda} \mathbf{S}_{\mathbf{R}, \mathbf{R}'} \mathbf{S}'^{-\lambda} \mathbf{S}_{\mathbf{R}, \mathbf{R}'}^{\top}) \right)^{1/2}. \quad (32)$$

The hyperparameter $\lambda \geq 0$ corresponds to the regularization weight of the ridge regression models over the representations. For all $\lambda \geq 0$, GULP is unbounded, satisfies the properties of a distance metric, and is invariant to orthogonal transformations, scaling, and translations. For $\lambda = 0$, GULP is invariant to affine transformations and can further be expressed as a linear transformation of the mean-squared CCA measure (Equation (12)).

Transferred Discrepancy (TD) [41] used an approach similar to GULP by measuring the discrepancy of linear classifiers, instead of linear regression models. TD is also related to the mean-squared CCA measure (Equation (12)).

Riemannian Distance. This measure considers the special geometry of matrices, which lie on a Riemannian manifold [12]. Every inner product defined on a Riemannian manifold induces a distance metric that considers the special curvature of these structures. On the manifold of symmetric positive definite matrices,

$$m_{\text{Riemann}}(\mathbf{R}, \mathbf{R}') = \sqrt{\sum_{i=1}^N \log^2(\lambda_i)}, \quad (33)$$

denotes such a metric, where λ_i is the i th eigenvalue of $\mathbf{S}^{-1} \mathbf{S}'$. Shahbazi et al. [125] proposed this measure using RSMs defined as $\mathbf{S} = \mathbf{R} \mathbf{R}^{\top} / D$. This matrix, however, can only be positive definite if $D > N$, which limits applicability of this measure. This measure is invariant to orthogonal transformations. Equivalence is indicated by a value of zero and larger values indicate dissimilarity.

Relational Knowledge Loss. A common approach to transfer knowledge in the context of knowledge distillation is to train the student model to mimic the relations between the teacher's instance representations [49]. This is done by minimizing the total element-wise difference of RSMs with respect to a loss function $l : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}_+$,

$$m_{\text{RK}}(\mathbf{R}, \mathbf{R}') = \sum_{i,j=1}^N l(\mathbf{S}_{i,j}, \mathbf{S}'_{i,j}). \quad (34)$$

While we defined RSMs via pairwise similarities, this approach has notably been generalized to higher-dimensional RSMs. For example, Park et al. [112] considered three-dimensional RSMs $\mathbf{S} \in \mathbb{R}^{N \times N \times N}$, where each entry $\mathbf{S}_{i,j,k}$ corresponds to the cosine of the angle enclosed by the vectors $\mathbf{v}_{i,j} = \mathbf{R}_i - \mathbf{R}_j$ and $\mathbf{v}_{k,j} = \mathbf{R}_k - \mathbf{R}_j$, i.e., $\mathbf{S}_{i,j,k} = \cos\text{-sim}(\mathbf{v}_{i,j}, \mathbf{v}_{j,k})$. Similarities are then aggregated over all instance triples. This measure instantiation is invariant to orthogonal transformations, translations, and scaling. Equivalence is indicated by a value of zero, larger values indicate dissimilarity.

3.4 Neighborhood-based Measures

The measures in this section compare the nearest neighbors of instances in the representation space. More precisely, each of these measures determine the k nearest neighbors of each instance representation \mathbf{R}_i in the full representation matrix \mathbf{R} with respect to a given similarity function s . In that context, the neighborhood size k is a parameter that has to be chosen for the application at hand. Letting \mathbf{S} denote the RSM of representation \mathbf{R} , and w.l.o.g. assuming that higher values indicate more similar representations, we formally define the set of the k nearest neighbors of the instance representation \mathbf{R}_i as the set $\mathcal{N}_{\mathbf{R}}^k(i) \subset \{j : 1 \leq j \leq N, j \neq i\}$ with $|\mathcal{N}_{\mathbf{R}}^k(i)| = k$ for which it holds that $\mathbf{S}_{i,j} > \mathbf{S}_{i,l}$ for all $j \in \mathcal{N}_{\mathbf{R}}^k(i), l \notin \mathcal{N}_{\mathbf{R}}^k(i) \cup \{i\}$. Once the nearest neighbors sets are determined, they are either compared directly, or one further considers distances of the representation \mathbf{R}_i to its nearest neighbors. For each of these measures, we then obtain a vector of instancewise neighborhood similarities $(v_{\text{NN-sim}}^k(\mathbf{R}, \mathbf{R}')_i)_{i \in \{1, \dots, N\}}$, which are averaged over all

instances to obtain similarity measures for the full representations \mathbf{R}, \mathbf{R}' :

$$m_{\text{NN-sim}}^k(\mathbf{R}, \mathbf{R}') = \frac{1}{N} \sum_{i=1}^N v_{\text{NN-sim}}^k(\mathbf{R}, \mathbf{R}')_i. \quad (35)$$

However, the instancewise similarities and their distribution could also be inspected more closely to obtain additional insights [71]. For brevity, in all the measures that we introduce in the following, we only give a description of how the instancewise similarities are computed. Similarly to RSM-based measures, the choice of the similarity function s determines which transformations these measures are invariant to. By default, and in line with the literature, we assume the use of cosine similarity (Equation (A.5)), which leads to invariance to orthogonal transformations and isotropic scaling.

k -NN Jaccard Similarity. This measure, also named *Nearest Neighbor Graph Similarity* [52] and *Nearest Neighbor Topological Similarity* [61], considers how many of the k nearest neighbors each instance has in common over a given pair of representations. The instancewise neighborhood similarities are computed in terms of the Jaccard similarities of the neighborhood sets $\mathcal{N}_R^k(i), \mathcal{N}_{R'}^k(i)$:

$$(v_{\text{jac}}^k(\mathbf{R}, \mathbf{R}'))_i := \frac{|\mathcal{N}_R^k(i) \cap \mathcal{N}_{R'}^k(i)|}{|\mathcal{N}_R^k(i) \cup \mathcal{N}_{R'}^k(i)|}. \quad (36)$$

Jaccard similarity is bounded in the interval $[0, 1]$, with a value of one indicating identical neighborhoods. Aside from the commonly used cosine similarity [123, 149], Euclidean distance was also used as similarity function [61].

Second-order Cosine Similarity. This measure was proposed by Hamilton et al. [53] to analyze changes in word embeddings over time. For each instance i , it first computes the union of nearest neighbors as an ordered set $\{j_1, \dots, j_{K(i)}\} := \mathcal{N}_R^k(i) \cup \mathcal{N}_{R'}^k(i)$ in \mathbf{R} and \mathbf{R}' in terms of cosine similarity (Equation (A.5)). Then the cosine similarities to these neighbors are compared between the two representations. Utilizing the cosine similarity RSMs \mathbf{S}, \mathbf{S}' of the representations \mathbf{R}, \mathbf{R}' , the instancewise second-order cosine similarities can then be defined as follows:

$$(v_{\text{2nd-cos}}^k(\mathbf{R}, \mathbf{R}'))_i := \text{cos-sim}((S_{i,j_1}, \dots, S_{i,j_{K(i)}}), (S'_{i,j_1}, \dots, S'_{i,j_{K(i)}})).$$

This measure is bounded in the interval $[0, 1]$, with $m_{\text{2nd-cos}}^k(\mathbf{R}, \mathbf{R}') = 1$ indicating equivalence of \mathbf{R} and \mathbf{R}' .

Rather than considering the union of the neighborhood sets, Chen et al. [24] considered the intersection of the top- k neighborhoods. Another similar approach was presented by Moschella et al. [102], who used a random fixed set of reference instances instead of neighbors. Further, Pointwise Normalized Kernel Alignment [71] can be seen as a variant of second-order cosine similarity with $k = N$ but different similarity function s for the RSM.

Rank Similarity. The k -NN Jaccard similarity captures the extent to which two neighborhood sets overlap but not the order of the common neighbors within those sets. To increase the importance of close neighbors, Wang et al. [149] determined distance-based ranks $r_{R_i}(j)$ to all $j \in \mathcal{N}_R^k(i)$, where $r_{R_i}(j) = n$ if R_j is the n th closest neighbor of R_i with respect to a given similarity function s . Based on these ranks, they defined the instance-based similarities as

$$(v_{\text{ranksim}}^k(\mathbf{R}, \mathbf{R}'))_i = \frac{1}{(\mathbf{v}_{\text{max}})_i} \cdot \sum_{j \in \mathcal{N}_R^k(i) \cap \mathcal{N}_{R'}^k(i)} \frac{2}{(1+|r_{R_i}(j)-r_{R'_i}(j)|)(r_{R_i}(j)+r_{R'_i}(j))}, \quad (37)$$

where $(\mathbf{v}_{\text{max}})_i = \sum_{k=1}^K \frac{1}{k}$, with $K = |\mathcal{N}_R^k(i) \cap \mathcal{N}_{R'}^k(i)|$, is a normalization factor that limits the maximum of the ranking similarity to one, which is achieved for completely identical rankings. Intuitively, the first factor of the denominator in Equation (37) measures the similarity of the ranks of an instance, whereas the second factor assigns rank-based weights to this similarity, with lower-ranked instances gaining less influence.

Joint Rank and k-NN Jaccard Similarity. Rank similarity has the issue that it is only calculated on the intersection of the k -nearest neighbor sets in different representations. That means rank similarity might be high, even if the k -NN sets have almost no overlap. Similarly, Jaccard similarity might be high, but the order of the nearest neighbors might be completely different. Therefore, Wang et al. [149] combined these two approaches to calculate the *Embedding Stability* by considering the product of Jaccard and rank similarity. Thus, using the instance vectors defined in Equations (36) and (37), we can define the vector of instancewise similarities as

$$(\mathbf{v}_{\text{Jac-Rank}}^k(\mathbf{R}, \mathbf{R}'))_i = (\mathbf{v}_{\text{Jac}}^k(\mathbf{R}, \mathbf{R}'))_i \cdot (\mathbf{v}_{\text{ranksim}}^k(\mathbf{R}, \mathbf{R}'))_i. \quad (38)$$

Scores are bounded in the interval $[0, 1]$, with $m_{\text{Jac-Rank}}^k(\mathbf{R}, \mathbf{R}') = 1$ indicating perfect similarity.

3.5 Topology-based Measures

The measures in this category are motivated by the *manifold hypothesis* [48, Section 5.11.3], which states that high-dimensional representations are expected to be concentrated in the vicinity of a low-dimensional data manifold \mathcal{M} . Following this assumption, these measures then aim to approximate the manifolds in terms of discrete topological structures such as graphs, or, more generally, (abstract) simplicial complexes [56], based on which the representations can then be compared. A simplicial complex can be seen as a generalization of graphs, in which vertices may not only be paired by edges but also can form higher-dimensional simplices. In both cases, each instance i typically corresponds to a vertex $v_i \in \mathcal{V}$, and edges/simplices are formed from instances that are close together in the representation space.

Geometry Score. The **Geometry Score (GS)** [66] characterizes representations by the number of *one-dimensional holes* in their data manifolds. To obtain this number of holes, the manifold is approximated in terms of simplicial complexes \mathcal{S}_α , $\alpha > 0$, in which vertices v_i form a simplex, if the α -neighborhoods of their representations \mathbf{R}_i overlap with each other. On this simplicial complex, the number of one-dimensional holes corresponds to the number of specific cycles in the complex and can be efficiently computed as the rank of its first homology group H_1 .

Given that the number of holes may differ dependent on α , and that there is no ground-truth regarding which value of α yields the most accurate approximation of the manifold, Khrulkov and Oseledets [66] suggest varying the value α between 0 and $\alpha_{\max} \propto \max_{i,j} \|\mathbf{R}_i - \mathbf{R}_j\|_2$. For each number of holes k , they collect the longest intervals (α_1, α_2) , in which the number of holes is constant at k , into sets \mathcal{B}_k . Then, the *relative living time* of k holes, which is defined as $\text{RLT}(k, \mathbf{R}) = \frac{1}{\alpha_{\max}} \sum_{(\alpha_1, \alpha_2) \in \mathcal{B}_k} (\alpha_2 - \alpha_1)$ can be considered as the probability that k holes exist in the manifold. Since building simplicial complexes from large data is computationally challenging, Khrulkov and Oseledets [66] suggested sampling numerous subsets \mathcal{I} of $n < N$ instances to build multiple so-called witness complexes with much lower number of simplices. Finally, one then considers the *mean relative living times (MRLT)* resulting from these complexes:

$$m_{\text{GS}}(\mathbf{R}, \mathbf{R}') = \sum_{k=0}^{k_{\max}-1} (\text{MRLT}(k, \mathbf{R}) - \text{MRLT}(k, \mathbf{R}'))^2, \quad (39)$$

where k_{\max} denotes the maximum number of holes that is considered. The authors suggested using $k_{\max} = 100$, aggregating the RLTs from 10,000 complexes of $n = 64$ vertices each, and setting $\alpha_{\max} = \frac{1}{128} \cdot \frac{N}{5000} \cdot \max_{i,j \in \mathcal{I}} \|\mathbf{R}_i - \mathbf{R}_j\|_2$ for each sample \mathcal{I} . This measure is bounded in the interval $[0, k_{\max}]$, with $m_{\text{GS}}(\mathbf{R}, \mathbf{R}') = 0$ indicating equivalent representations. Further, it is invariant to orthogonal transformations, isotropic scaling, and translations.

Multi-Scale Intrinsic Distance. The *Multi-Scale Intrinsic Distance (IMD)* [144] applies k -NN graphs $\mathcal{G}(\mathbf{R})$ as a proxy to characterize and compare the manifold of the representations. Specifically, Tsitsulin et al. [144] utilize the *heat kernel trace* on $\mathcal{G}(\mathbf{R})$ to compare representations, which is defined as $\text{hkt}_{\mathcal{G}(\mathbf{R})}(t) = \sum_i e^{-t\lambda_i}$, with λ_i as the eigenvalues of the normalized graph Laplacian

of $\mathcal{G}(\mathbf{R})$. Similarity between the manifolds is then computed as a lower bound of the Gromov–Wasserstein distance, which can be expressed in terms of the heat kernel trace:

$$m_{\text{IMD}}(\mathbf{R}, \mathbf{R}') = \sup_{t>0} e^{-2(t+t^{-1})} |\text{hkt}_{\mathcal{G}(\mathbf{R})}(t) - \text{hkt}_{\mathcal{G}(\mathbf{R}')} (t)|. \quad (40)$$

Practically, Tsitsulin et al. [144] approximate $\text{hkt}_{\mathcal{G}(\mathbf{R})}(t)$ using the *Stochastic Lanczos Quadrature* [145] and obtain the supremum by sampling t from a parameter grid. They built the graph using the $k = 5$ nearest neighbors with respect to Euclidean distance. The IMD has no upper bound; the minimum, which indicates maximal similarity, is zero. It is invariant to orthogonal transformations, isotropic scaling and translations.

Representation Topology Divergence. Similarly to the geometry score, *Representation Topology Divergence (RTD)* [8] also considers persistence intervals of topological features of representations. However, in this approach graphs are applied for simplicial approximation of the representations, and the number of their connected components are the topological feature of interest. Specifically, Barannikov et al. [8] first compute RSMs with Euclidean distance and normalize these by the 90th percentile of their values. Then, for a given distance threshold $\alpha > 0$, they construct a graph $\mathcal{G}^\alpha(\mathbf{R})$ with its adjacency matrix \mathbf{A} defined as $A_{i,j} = S_{i,j} \cdot \mathbb{1}\{S_{i,j} < \alpha\}$, and a union graph $\mathcal{G}^\alpha(\mathbf{R}, \mathbf{R}')$ with its adjacency matrix \mathbf{A} defined as $A_{i,j} = \min(S_{i,j}, S'_{i,j}) \cdot \mathbb{1}\{\min(S_{i,j}, S'_{i,j}) < \alpha\}$. If $\mathcal{G}^\alpha(\mathbf{R})$ and $\mathcal{G}^\alpha(\mathbf{R}, \mathbf{R}')$ differ in the number of their connected components, then this is considered a topological discrepancy. For each specific discrepancy that occurs for varying values of α , the longest corresponding interval (α_1, α_2) , for which this discrepancy persists, is collected in a set $\mathcal{B}(\mathbf{R}, \mathbf{R}')$. The total length of these intervals, denoted as $b(\mathbf{R}, \mathbf{R}') = \sum_{(\alpha_1, \alpha_2) \in \mathcal{B}(\mathbf{R}, \mathbf{R}')} \alpha_2 - \alpha_1$, then quantifies similarity between two representations. The final RTD measure is constructed by subsampling K subsets $\mathcal{I}^{(k)}$ of $n < N$ instances each, and collecting the values $b(\mathbf{R}^{(k)}, \mathbf{R}'^{(k)})$ derived from the representations $\mathbf{R}^{(k)} = (\mathbf{R}_i)_{i \in \mathcal{I}^{(k)}} \in \mathbb{R}^{n \times D}$ to form a measure $\text{RTD}(\mathbf{R}, \mathbf{R}') = \frac{1}{K} \sum_{i=1}^K b(\mathbf{R}^{(k)}, \mathbf{R}'^{(k)})$. Because RTD is asymmetric, the authors proposed to use

$$m_{\text{RTD}}(\mathbf{R}, \mathbf{R}') = \frac{1}{2}(\text{RTD}(\mathbf{R}, \mathbf{R}') + \text{RTD}(\mathbf{R}', \mathbf{R})). \quad (41)$$

For hyperparameters, they suggested using $K = 10$ subsets of $n = 500$ representations each as default values. An RTD of zero indicates equivalent representations, with higher values indicating less similarity. By construction of the RSMs, RTD is invariant to orthogonal transformations, isotropic scaling, and translations.

3.6 Descriptive Statistics

Measures of this category deviate from all previous measures in a way that they describe statistical properties of either (i) individual representations \mathbf{R} , or (ii) measures of variance in the instance representations \mathbf{R}_i over sets \mathcal{R} of more than two representations. In case of (i), the statistics can be directly compared over pairs or sets of representations. For case (ii), one could aggregate or analyze the distribution of the instancewise variations. While there are numerous statistics that could be used to compare representations, in the following we specifically outline statistics that have already been used to characterize representations in existing literature.

Intrinsic Dimension. The *intrinsic dimension* of a representation \mathbf{R} corresponds to the minimal number of variables that are necessary to describe its data points. It can be defined as the lowest value $M \in \mathbb{N}, M < N$, for which the representation \mathbf{R} lies in a M -dimensional manifold of \mathbb{R}^N [21]. This statistic has its roots in social sciences [127] and information theory [11] and has since been applied in countless other fields, resulting in different variants and numerous methods to estimate its exact values—for more details, we point the interested reader to the survey by Camastra and Staiano [21]. In the context of neural network analysis, different variants of the intrinsic dimension

have been used as a tool to analyze the amount of information that is contained and processed within high-dimensional layers [2, 9, 88]. This statistic is invariant to affine transformations.

Magnitude. Wang et al. [149] characterized *magnitude* as the Euclidean length of instance representations R_i . Consequently, they considered the length of the mean instance representation as a statistic for a representation R ,

$$m_{\text{Mag}}(R) := \left\| \frac{1}{N} \sum_{i=1}^N R_i \right\|_2. \quad (42)$$

Aside from aggregating magnitude over all instances, they further proposed a measure to quantify the variance of the magnitude of instancewise representations over multiple models. More precisely, given a set of representations \mathcal{R} , Wang et al. [149] measured the variance in the magnitudes of individual instances i as

$$m_{\text{Var-Mag}}(\mathcal{R}, i) = \frac{1}{\max_{R \in \mathcal{R}} \|R_i\|_2 - \min_{R \in \mathcal{R}} \|R_i\|_2} \cdot \sqrt{\frac{1}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} (\|R_i\|_2 - \bar{d}_i(\mathcal{R}))^2}, \quad (43)$$

where $\bar{d}_i(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} \|R_i\|_2$ is the average magnitude of the representations of instance i in \mathcal{R} . As magnitude is unaffected by transformations that preserve vector length, this statistic is invariant to orthogonal transformations.

Concentricity. Wang et al. [149] proposed *concentricity* as a measure of the density of representations. It is based on measuring the cosine similarities of each instance representation R_i to the average representation, which we denote as $\alpha_i(R) = \cos\text{-sim}(R_i, \frac{1}{N} \sum_{j=1}^N R_j)$. Similarly to magnitude, Wang et al. [149] then considered the mean concentricity

$$m_{\text{mConc}}(R) := \frac{1}{N} \sum_{i=1}^N \alpha_i(R) \quad (44)$$

as a statistic for a single model and measured the instancewise variance of concentricity via

$$m_{\text{Var-Conc}}(\mathcal{R}, i) = \frac{1}{\max_{R \in \mathcal{R}} \alpha_i(R) - \min_{R \in \mathcal{R}} \alpha_i(R)} \cdot \sqrt{\frac{1}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} (\alpha_i(R) - \bar{\alpha}_i(\mathcal{R}))^2}, \quad (45)$$

where $\bar{\alpha}_i(\mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{R \in \mathcal{R}} \alpha_i(R)$ is the average concentricity of instance i in \mathcal{R} . Concentricity inherits from cosine similarity the invariances to orthogonal transformations and isotropic scaling.

Uniformity. *Uniformity* [52, 153] quantifies density of representations by measuring how close the distribution of instance representations is to a uniform distribution on the unit hypersphere. This measure is defined as

$$m_{\text{uniformity}}(R) = \log \left(\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N e^{-t \|R_i - R_j\|_2^2} \right), \quad (46)$$

where t is a hyperparameter that was set to $t = 2$ by Wang et al. [151] and Gwilliam and Shrivastava [52]. The statistic is bounded in the interval $[0, 1]$, with $m_{\text{uniformity}}(R) = 1$ indicating perfectly uniform representations. Uniformity is invariant to orthogonal transformations and translation, as these transformations preserve distances.

Tolerance. This statistic considers the proximity of representations of semantically similar inputs [150]. In contrast to the previous statistics, it requires a vector of ground-truth labels $\mathbf{y} \in \mathbb{R}^N$. Further, it is assumed that all instance representations have unit norm. *Tolerance* is computed as the mean similarity of inputs with the same class:

$$m_{\text{tol}}(R) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (R_i^\top R_j) \cdot \mathbb{1}\{\mathbf{y}_i = \mathbf{y}_j\}. \quad (47)$$

Tolerance is bounded in the interval $[-1, 1]$, with $m_{\text{tol}}(R) = 0$ indicating that representations that share the same label are always uncorrelated. This statistic is invariant to orthogonal transformations and isotropic scaling.

Instance-Graph Modularity. Similarly to the topology-based measures (see Section 3.5), Saini et al. [122] and Lu et al. [87] proposed measures based on building a graph to model representations, though this was not motivated from a topological perspective. Specifically, they used *modularity*

[106] to identify whether semantically similar inputs are close together in the graph, and consequently, the representation space. In both cases, a sparse graph was constructed. Saini et al. [122] used the affinity matrix resulting from sparse subspace clustering [38] as the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, whereas Lu et al. [87] determined the adjacency matrix element-wise via $\mathbf{A}_{i,j} = \mathbf{S}_{i,j} \cdot \mathbb{1}\{j \in \mathcal{N}_R^k(i)\}$, considering a cosine similarity-based RSM \mathbf{S} . The modularity of the network, and in consequence the statistic for \mathbf{R} , is then defined as

$$m_{\text{Mod}}(\mathbf{R}) = \frac{1}{2W} \sum_{i,j} \left(\mathbf{A}_{i,j} - \frac{d_i d_j}{W} \right) \cdot \mathbb{1}\{\mathbf{y}_i = \mathbf{y}_j\}, \quad (48)$$

where $d_i = \sum_j \mathbf{A}_{i,j}$ denotes the effective degree of node v_i , $W = \sum_{i,j} \mathbf{A}_{i,j}$ is a normalization factor, and \mathbf{y} is the vector of ground-truth labels. The maximum modularity is given by 1, and high modularity implies that nodes of the same label are highly connected with each other, with only few connections to nodes of another label. Both variants are invariant to orthogonal transformations. The variant by Lu et al. [87] is additionally invariant to isotropic scaling.

Neuron-Graph Modularity. Lange et al. [77] also considered modularity as a statistic to characterize representations. However, in their approach, the nodes v_j represented neurons $\mathbf{R}_{-,j}$ instead of instance representations \mathbf{R}_i . To model the similarity of neurons that is needed to construct the graphs, they proposed four different variants of RSMs that either consider pure neuron activations or also gradients with respect to neuron activations. In that latter case, one may consider the modularity based on such RSMs as a hybrid measure of representational and functional characteristics.

Once an RSM $\mathbf{S} \in \mathbb{R}^{D \times D}$ has been computed, Lange et al. [77] constructed the adjacency matrix \mathbf{A} of $\mathcal{G}(\mathbf{R})$ via $\mathbf{A}_{i,j} = \mathbf{S}_{i,j} \cdot (1 - \mathbb{1}\{i = j\})$. Unlike Lu et al. [87], they did not allocate nodes to clusters based on ground-truth labels but determined an optimal soft assignment of n clusters that maximizes modularity. Specifically, they tried to find an optimal cluster assignment matrix $\mathbf{C} \in \mathbb{R}^{D \times n}$, where each entry $C_{j,k} \in [0, 1]$ determines the assignment of neuron j to cluster k . The number of clusters $n \leq D$ of neuron activations is a parameter that is to be optimized as well. Given a definition of clustering from Girvan and Newman [45], neuron modularity is then defined as

$$m_{\text{nMod}}(\mathbf{R}) = \max_{\mathbf{C}} \text{tr}(\mathbf{C}^T \tilde{\mathbf{A}} \mathbf{C}) - \text{tr}(\mathbf{C}^T \mathbf{1}_D^T \mathbf{1}_D \tilde{\mathbf{A}} \mathbf{C}), \quad (49)$$

where $\tilde{\mathbf{A}} = \frac{1}{\mathbf{1}_D^T \mathbf{A} \mathbf{1}_D} \mathbf{A}$ is the normalized adjacency matrix. To determine the cluster assignment \mathbf{C} , they provided an approximation method based on Newman's modularity maximization algorithm [105]. Generally, m_{nMod} is invariant to permutations, since these effectively only relabel the nodes in the resulting graph.

4 Functional Similarity Measures

Next, we present functional similarity measures. As mentioned in Section 2.2, these measures compare outputs $\mathbf{O}, \mathbf{O}' \in \mathbb{R}^{N \times C}$, where each element $\mathbf{O}_{i,c}$ denotes the probabilities or scores of class c for input X_i and $\arg \max_c \mathbf{O}_{i,c} = \hat{c}$ indicates that class \hat{c} is the prediction for input X_i . We mainly categorize measures based on the granularity of the model outputs that they require, as illustrated in Figure 4. An overview of all measures is given in Table 2.

4.1 Performance-based Measures

A popular view on functional similarity is that models are similar if they reach similar performance on some downstream task (e.g., References [7, 30, 34, 80]). This approach is easy to implement, as the comparison of models is reduced to comparing two scalar performance scores, such as accuracy. However, this simplification also obfuscates more nuanced differences in functional behavior, which cannot directly be captured with a single number per model.

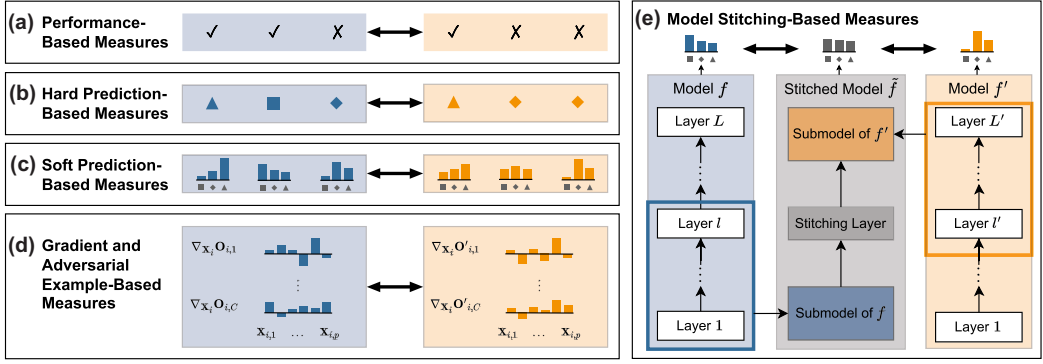


Fig. 4. Types of functional similarity measures, illustrated in the context of classifying inputs with respect to their shape (\diamond , \square , \triangle). Performance-based (a), hard prediction-based (b), soft prediction-based (c), and gradient and adversarial example-based measures (d) compare outputs of different granularity. Model stitching (e) combines parts of two models and measures functional similarity between the resulting model and the original models.

Most commonly, given some quality function q that evaluates the performance of a model with respect to ground-truth labels, the (absolute) difference in performance is used for similarity:

$$m_{\text{Perf}}(\mathbf{O}, \mathbf{O}') = |q(\mathbf{O}) - q(\mathbf{O}')|. \quad (50)$$

Although accuracy is an often used quality function in the literature [7, 30, 34, 80], other performance metrics such as F1 score may be used [158]. However, choosing performance metrics that capture relevant aspects of functional behavior requires careful consideration [118].

4.2 Hard Prediction-based Measures

The measures in this section quantify functional similarity by comparing hard predictions. Thus, each measure of this category will report high similarity if the hard predictions agree for most inputs, regardless of correctness or confidence. These measures are related to literature on ensemble diversity [76, 140] and inter-rater agreement [6, 143].

Disagreement. *Disagreement*, also known as *churn* [39], *jitter* [86], or *Hamming prediction differences* [126], is the expected rate of conflicting hard predictions over inputs and models [90, 131]. Due to its simplicity and interpretability, it is a particularly popular measure for functional similarity. Formally, disagreement between two models is defined as

$$m_{\text{Dis}}(\mathbf{O}, \mathbf{O}') = \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{\arg \max_j \mathbf{O}_{i,j} \neq \arg \max_j \mathbf{O}'_{i,j}\}. \quad (51)$$

The measure is bounded in the interval $[0, 1]$, with a score of zero indicating perfect agreement, and a score of one indicating completely distinct functional behavior. Practically, this range is bounded by model quality, with high disagreement being impossible if the compared models are both very accurate. Further, there are bounds on disagreement that depend on the soft predictions of the compared models [14].

Error-corrected Disagreement. As the range of possible disagreement values depends on the accuracy of the compared models, Fort et al. [43] proposed to correct for this influence by dividing the disagreement by the error rate $q_{\text{Err}}(\mathbf{O}) := q_{\text{Err}}(\mathbf{O}, \mathbf{y}) := \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{\arg \max_j \mathbf{O}_{i,j} \neq \mathbf{y}_i\}$ of one of

Table 2. Overview of Functional Similarity Measures

| Type | Measure | Groupwise | Blackbox Access | Labels Required | Similarity ↑ |
|----------------------------|---|-----------|-----------------|-----------------|--------------|
| Performance | Performance Difference | ✗ | ✓ | ✓ | ✗ |
| Hard Prediction | Disagreement [39, 86, 90, 126] | ✗ | ✓ | ✗ | ✗ |
| | Error-Corrected Disagreement [43] | ✗ | ✓ | ✓ | ✗ |
| | Min-Max-normalized Disagreement [69] | ✗ | ✓ | ✓ | ✗ |
| | Kappa Statistic [27] | ✗* | ✓ | ✗ | ✓ |
| | Ambiguity [93, 123] | ✓ | ✓ | ✗ | ✗ |
| | Discrepancy [93] | ✓ | ✓ | ✗ | ✗ |
| | Label Entropy [31] | ✓ | ✓ | ✗ | ✗ |
| Soft Prediction | Norm of Soft Prediction Difference [4, 163] | ✗ | ✓ | ✗ | ✗ |
| | Surrogate Churn [14] | ✗ | ✓ | ✗ | ✗ |
| | Jensen-Shannon Divergence [85] | ✗ | ✓ | ✗ | ✗ |
| | Prediction Difference [126] | ✓ | ✓ | ✗ | ✗ |
| | Rashomon Capacity [62] | ✓ | ✓ | ✗ | ✗ |
| Gradient & Adversarial Ex. | ModelDiff [82] | ✗ | ✗‡ | ✓ | ✓ |
| | Adversarial Transferability [63] | ✗ | ✗‡ | ✓ | ✓ |
| | Saliency Map Similarity [64] | ✗ | ✗ | ✗ | ✓ |
| Stitching | Performance Difference [7, 30, 80] | ✗ | ✗ | ✓ | ✗† |

*: Groupwise variants available. †: Depends on comparison. ‡: Depends on adversarial example generation.

We indicate whether measure enable groupwise comparison of models, whether they can be applied with blackbox access to the models, and if they require ground-truth labels. *Similarity* ↑ indicates whether increasing scores imply increasing similarity of models.

the models:

$$m_{\text{ErrCorrDis}}(\mathbf{O}, \mathbf{O}') = \frac{m_{\text{Dis}}(\mathbf{O}, \mathbf{O}')}{q_{\text{Err}}(\mathbf{O})}. \quad (52)$$

By design, this measure is not symmetric, since the error rates of the outputs \mathbf{O}, \mathbf{O}' may vary. A normalized disagreement of zero indicates perfect agreement, whereas the upper limit is dependent on the error rate—exact limits are provided by Fort et al. [43], which help to contextualize the similarity scores that are obtained.

A normalized and symmetric variant of this measure was used by Klabunde and Lemmerich [69]. Their *Min-Max-normalized disagreement* measure relates the observed disagreement $m_{\text{Dis}}(\mathbf{O}, \mathbf{O}')$ to the minimum and maximum possible disagreement, given error rates $q_{\text{Err}}(\cdot)$ of the models. The minimum is computed as $m_{\text{Dis}}^{(\min)}(\mathbf{O}, \mathbf{O}') = |q_{\text{Err}}(\mathbf{O}) - q_{\text{Err}}(\mathbf{O}')|$, and the maximum possible disagreement as $m_{\text{Dis}}^{(\max)}(\mathbf{O}, \mathbf{O}') = \min(q_{\text{Err}}(\mathbf{O}) + q_{\text{Err}}(\mathbf{O}'), 1)$, leading to the measure

$$m_{\text{MinMaxNormDis}}(\mathbf{O}, \mathbf{O}') = \frac{m_{\text{Dis}}(\mathbf{O}, \mathbf{O}') - m_{\text{Dis}}^{(\min)}(\mathbf{O}, \mathbf{O}')}{m_{\text{Dis}}^{(\max)}(\mathbf{O}, \mathbf{O}') - m_{\text{Dis}}^{(\min)}(\mathbf{O}, \mathbf{O}')}. \quad (53)$$

This measure is bounded in the interval $[0, 1]$, with $m_{\text{MinMaxNormDis}}(\mathbf{O}, \mathbf{O}') = 0$ indicating perfect agreement.

Chance-corrected Disagreement. Rather than correcting for accuracy of models, one can correct for the rate of agreement that two or more classification models are expected to have by chance. The probably most prominent measure that follows this rationale is *Cohen's kappa* [27], which was proposed as a measure for inter-rater agreement, but has also been used in machine learning [22, 44]. Assuming that the compared outputs \mathbf{O}, \mathbf{O}' are statistically independent, and letting k_c denote the absolute amount of times that class c is predicted in the output \mathbf{O} , the expected agreement rate of such models is given by $p_e = \frac{1}{N^2} \sum_{c=1}^C k_c k'_c$. Based on these values, Cohen's Kappa is defined as

$$m_{\text{Cohen}}(\mathbf{O}, \mathbf{O}') = 1 - \frac{m_{\text{Dis}}(\mathbf{O}, \mathbf{O}')}{1 - p_e} = \frac{p_o - p_e}{1 - p_e}, \quad (54)$$

where $p_o = 1 - m_{\text{Dis}}(\mathbf{O}, \mathbf{O}')$ denotes the observed agreement. When $m_{\text{Cohen}}(\mathbf{O}, \mathbf{O}') = 1$, perfect agreement of the models is indicated; a value $m_{\text{Cohen}}(\mathbf{O}, \mathbf{O}') < 0$ indicates less agreement than expected by chance.

To measure similarity between bigger sets of outputs, Fleiss's kappa [42] can be used as a more general variant [36]. The literature on inter-rater agreement [40] lists related measures that are more general or have weaker assumptions.

Groupwise Disagreement. Disagreement cannot identify commonalities across a whole set of models, as pairwise similarity of models does not imply groupwise similarity. The two following measures extend disagreement to identify functional similarity across sets of models.

First, *ambiguity* [93], also called *linear prediction overlap* [52], is the share of instances that receive conflicting predictions by *any* pair of models out of a given set of models. Ambiguity is defined as

$$m_{\text{Ambiguity}}(\mathbf{O}) = \frac{1}{N} \sum_{i=1}^N \max_{\substack{\mathbf{O}, \mathbf{O}' \in \mathcal{O} \\ \text{s.t. } \mathbf{O} \neq \mathbf{O}'}} \mathbb{1}\{\arg \max_j \mathbf{O}_{i,j} \neq \arg \max_j \mathbf{O}'_{i,j}\}. \quad (55)$$

The counterpart to ambiguity is the *stable core* measure proposed by Schumacher et al. [123], which counts the share of instances with consistent predictions. They also considered a relaxation of this consistency, in which an instance is only required to obtain the same prediction by a fixed proportion of models (e.g., 90% of all models) to be considered stable.

Second, *discrepancy* [93] gives the maximum disagreement between two classifiers from a set of multiple models:

$$m_{\text{Discrepancy}}(\mathbf{O}) = \max_{\substack{\mathbf{O}, \mathbf{O}' \in \mathcal{O} \\ \text{s.t. } \mathbf{O} \neq \mathbf{O}'}} \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{\arg \max_j \mathbf{O}_{i,j} \neq \arg \max_j \mathbf{O}'_{i,j}\}. \quad (56)$$

Both ambiguity and discrepancy are bounded in the interval $[0, 1]$, with a value of zero indicating perfect agreement.

Label Entropy. Datta et al. [31] measured the variance in individual predictions over a group of outputs in terms of entropy. Letting $k_c^{(i)}$ denote the number of times that instance i is predicted as class c , *Label Entropy* (LE) is defined as

$$m_{\text{LE}}(\mathbf{O}, i) = \sum_{c=1}^C -\frac{k_c^{(i)}}{|\mathcal{O}|} \log \left(\frac{k_c^{(i)}}{|\mathcal{O}|} \right). \quad (57)$$

Label Entropy is bounded in the interval $[0, \log(C)]$, with $m_{\text{LE}}(\mathbf{O}, i) = 0$ indicating identical predictions.

4.3 Soft Prediction-based Measures

This group of measures compares soft predictions, such as classwise probabilities or scores from decision functions. Intuitively, this provides more nuance to the notion of similarity in outputs, since we can consider differences in confidence of individual predictions. The impact of confidence is specifically exemplified by cases where scores are close to the decision boundary. Even a minimal change in scores may cause a different classification in one case, whereas scores would need to change drastically for a different classification in another case.

Norm of Soft Prediction Difference. A direct way to generalize disagreement to soft predictions is to apply a norm $\|\cdot\|$ on instancewise differences in soft predictions and average this over all inputs, which yields a measure

$$m_{\text{PredNormDiff}}(\mathbf{O}, \mathbf{O}') = \frac{1}{2N} \sum_{i=1}^N \|\mathbf{O}_i - \mathbf{O}'_i\|, \quad (58)$$

that assigns a score of zero when outputs are equal. Ba and Caruana [4] and Zhang et al. [163] applied this measure using the Euclidean norm to compare logits and probabilities, respectively.

Surrogate Churn. Bhojanapalli et al. [14] proposed *surrogate churn* (SChurn) as a relaxed version of disagreement, that takes into account the distribution of the soft predictions. For $\alpha > 0$, it is defined as

$$m_{\text{SChurn}}^\alpha(\mathbf{O}, \mathbf{O}') = \frac{1}{2N} \sum_{i=1}^N \left\| \left(\frac{\mathbf{O}_i}{\max_c \mathbf{O}_{i,c}} \right)^\alpha - \left(\frac{\mathbf{O}'_i}{\max_c \mathbf{O}'_{i,c}} \right)^\alpha \right\|_1. \quad (59)$$

A value $m_{\text{SChurn}}^\alpha(\mathbf{O}, \mathbf{O}') = 0$ indicates perfect agreement of outputs. The authors showed that when $\alpha \rightarrow \infty$, this measure is equivalent to standard disagreement (cf. Section 4.2), and we use $\alpha = 1$ as the default value.

Divergence-based Measures. When soft predictions represent class probabilities, divergence measures for probability distributions can be used to evaluate the similarity of instance-level predictions. For example, *Kullback–Leibler* (KL) divergence is commonly used when training similar models in knowledge distillation [49]. When focusing on pure similarity assessment, a common choice is to apply the symmetric *Jensen–Shannon Divergence* (JSD) by averaging over all instances [36, 43, 156]. Letting $\text{KL}(\cdot \parallel \cdot)$ denote the Kullback–Leibler divergence, this measure is defined as

$$m_{\text{JSD}}(\mathbf{O}, \mathbf{O}') = \frac{1}{2N} \sum_{i=1}^N \text{KL}(\mathbf{O}_i \parallel \bar{\mathbf{O}}_i) + \text{KL}(\mathbf{O}'_i \parallel \bar{\mathbf{O}}_i), \quad (60)$$

with $\bar{\mathbf{O}} = \frac{\mathbf{O} + \mathbf{O}'}{2}$ denoting the average output. Equality of outputs is given when $m_{\text{JSD}}(\mathbf{O}, \mathbf{O}') = 0$, and higher values indicate dissimilarity. A similar approach to compare probabilistic outputs is given as Graph Explanation Faithfulness [1]. An overview of divergence measures that could be used has been given by Cha [23].

Prediction Difference. Shamir and Coviello [126] specifically considered differences in predictions over more than two models. Their *prediction difference* (PD) intuitively quantifies the variance in model predictions. Letting $\bar{\mathbf{O}} = \frac{1}{|\mathcal{O}|} \sum_{\mathbf{O} \in \mathcal{O}} \mathbf{O}$ denote the average output matrix, their standard prediction difference measure aggregates instancewise deviations from the average output in terms of a p -norm,

$$m_{\text{PD}}^p(\mathcal{O}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{|\mathcal{O}|} \sum_{\mathbf{O} \in \mathcal{O}} \|\mathbf{O}_i - \bar{\mathbf{O}}_i\|_p. \quad (61)$$

Shamir and Coviello [126] used $p = 1$ for interpretable differences of probability distributions. $m_{\text{PD}}^p(\mathcal{O}) = 0$ indicates identical outputs of all models. Higher PD indicates higher dissimilarity between the compared models.

Next to norm-based prediction difference, Shamir and Coviello [126] further proposed a variant of the PD that relates the variance in the outputs to their average magnitude and a variant that considers class labels \mathbf{y} if these are given.

Rashomon Capacity. Similarly to label entropy (Equation (57)), *Rashomon Capacity* (RC) [62] also applies concepts from information theory to measure multiplicity in predictions on individual instances. Formally, letting $P_{\mathcal{O}}$ denote a probability distribution over the set of outputs \mathcal{O} , and $\Delta_C = \{\mathbf{p} \in [0, 1]^C : \sum_{i=1}^C \mathbf{p}_i = 1\}$ the probability simplex, it considers the output spread $\inf_{\mathbf{p} \in \Delta_C} \mathbb{E}_{\mathbf{O} \sim P_{\mathcal{O}}} \text{KL}(\mathbf{O}_i \parallel \mathbf{p})$, where $\mathbf{p} \in \Delta_C$ is a reference distribution that is optimized to minimize distances to all outputs. The Rashomon Capacity is then defined via the *channel capacity*, which maximizes the output spread over all probability distributions over the outputs:

$$m_{\text{RC}}(\mathcal{O}, i) = 2^{\text{Capacity}(\mathcal{O}, i)}, \quad \text{with} \quad \text{Capacity}(\mathcal{O}, i) = \sup_{P_{\mathcal{O}}} \inf_{\mathbf{p} \in \Delta_C} \mathbb{E}_{\mathbf{O} \sim P_{\mathcal{O}}} \text{KL}(\mathbf{O}_i \parallel \mathbf{p}). \quad (62)$$

To approximate the Rashomon Capacity of an instance, Hsu and Calmon [62] suggested using the Blahut–Arimoto algorithm [3, 16]. A similarity measure over all instances can be obtained by aggregation, e.g., via the mean value.

It holds that $m_{\text{RC}}(\mathcal{O}, i) \in [1, C]$ with $m_{\text{RC}}(\mathcal{O}, i) = 1$ if and only if all outputs are identical and $m_{\text{RC}}(\mathcal{O}, i) = C$ if and only if every class is predicted once with perfect confidence. Further, the measure is monotonous, i.e., it holds that $m_{\text{RC}}(\mathcal{O}', i) \leq m_{\text{RC}}(\mathcal{O}, i)$ for all $\mathcal{O}' \subseteq \mathcal{O}$.

4.4 Gradient and Adversarial Example-based Measures

The measures in this section use model gradients to characterize similarity either directly or indirectly via adversarial examples. A core assumption of these measures is that similar models have similar gradients. This assumption also leads to transferability of adversarial attacks, i.e., the behavior of the compared models changes similarly when given an adversarial example computed for only one of the models.

ModelDiff. In their *ModelDiff* measure, Li et al. [82] used adversarial examples from perturbation attacks to characterize decision regions, which can then be compared across two models. Given a model f , they first created adversarial examples \tilde{X}_i for every input X_i by adding noise to these inputs that steer the model away from a correct prediction. Such examples can be determined by methods such as projected gradient descent [91]. The difference between the instancewise original soft predictions $O_i = f(X_i)$ and the predictions for the corresponding adversarial example $\tilde{O}_i = f(\tilde{X}_i)$ is then collected in a *decision distance vector* defined as $(\mathbf{v}_{\text{DDV}}(O, \tilde{O}))_i = \cos\text{-sim}(O_i, \tilde{O}_i)$. Finally, they quantified the difference between models via the difference in the DDVs, measured with cosine similarity:

$$m_{\text{ModelDiff}}(O, O') = \cos\text{-sim}(\mathbf{v}_{\text{DDV}}(O, \tilde{O}), \mathbf{v}_{\text{DDV}}(O', \tilde{O}')). \quad (63)$$

The outputs $\tilde{O}'_i = f'(\tilde{X}_i)$ are computed from the same adversarial examples \tilde{X}_i . A similarity score of one indicates equivalence of outputs. Since this measure uses adversarial examples of only one of the models, it is not symmetric.

Adversarial Transferability. Similarly to ModelDiff, Hwang et al. [63] measured the similarity of networks in terms of the transferability of adversarial attacks. Given two networks f, f' , for each input X_i that is predicted correctly by both networks, a pair of corresponding adversarial examples $\tilde{X}_i, \tilde{X}'_i$ is generated with projected gradient descent [91]. These adversarial examples are then fed into the opposite model, yielding outputs $\tilde{O}_i = f(\tilde{X}'_i)$ and $\tilde{O}'_i = f'(\tilde{X}_i)$, for which it is then determined how often both are incorrect. Thus, given the vector of ground-truth labels \mathbf{y} , and letting $\mathcal{X}_{\text{true}}$ denote the set of instances that were predicted correctly by both models, Hwang et al. [63] defined the measure

$$m_{\text{AdvTrans}}(\tilde{O}, \tilde{O}') = \log \left[\max \left\{ \epsilon, \frac{100}{2|\mathcal{X}_{\text{true}}|} \sum_{i \in \mathcal{X}_{\text{true}}} (\mathbb{1}(\arg \max_j \tilde{O}_{i,j} \neq \mathbf{y}_i) + \mathbb{1}(\arg \max_j \tilde{O}'_{i,j} \neq \mathbf{y}_i)) \right\} \right], \quad (64)$$

where $\epsilon > 0$ is introduced to avoid $\log(0)$. A value of $m_{\text{AdvTrans}}(\tilde{O}, \tilde{O}') = \log(100)$ indicates perfect model similarity, whereas $m_{\text{AdvTrans}}(\tilde{O}, \tilde{O}') = \log(\epsilon)$ indicates complete disagreement.

Cosine Similarity of Saliency Maps. Jones et al. [64] used a direct approach to compare models in terms of their gradients. They computed the cosine similarity between (vectorized) saliency maps [130], which model the impact of input features on individual predictions. Practically, this impact is quantified using instancewise gradients $\nabla_{X_i} O_{i,c}$, and the instancewise similarities then aggregated to yield the following measure:

$$m_{\text{SaliencyMap}}(O, O') = \frac{1}{nC} \sum_{i=1}^N \sum_{c=1}^C \cos\text{-sim}(|\nabla_{X_i} O_{i,c}|, |\nabla_{X_i} O'_{i,c}|), \quad (65)$$

where the absolute value $|\cdot|$ is applied element-wise (for inputs with a single channel). A value $m_{\text{SaliencyMap}}(O, O') = 1$ indicates perfect similarity, with lower values indicating stronger differences between models.

4.5 Stitching-based Measures

The intuition behind *stitching* is that similar models should be similar in their internal processes and, thus, swapping layers between such models should not result in big differences in the outputs if a layer that converts representations is introduced [7, 30, 80]. Given two models

f, f' , stitching consists of training a *stitching layer* (or network) g to convert representations from f at layer l into representations of f' at layer l' . One then considers the composed model $\tilde{f} := f^{(L')} \circ \dots \circ f^{(l'+1)} \circ g \circ f^{(l)} \circ f^{(l-1)} \circ \dots \circ f^{(1)}$, which uses the bottom-most layers of f and the top-most layers of f' , and compares its output with the original models. Most commonly they are compared in terms of a quality function q such as accuracy [7, 30]. This yields a measure

$$m_{\text{stitch}}(\tilde{O}, O') = q(\tilde{O}) - q(O'), \quad (66)$$

where $\tilde{O} = \tilde{f}(X)$ is the output of the stitched model. However, other functional similarity measures can also be used.

Both design and placement of stitching layers affects assessments of model similarity, and several types of stitching layers were studied [30, 47]. Bansal et al. [7] chose stitching layers such that the architecture of the stitched model is consistent with the original models. For instance, they use a tokenwise linear function to stitch transformer blocks. For CNNs, 1×1 convolutions are generally used in stitching layers [7, 30, 80].

Compared to other measures, model stitching requires training an additional layer and thus might be more costly to implement. Further, (non-deterministic) training of the stitching layer presents a source of instability of the final results. To train the stitching layers, one typically freezes parameters of the original models and only optimizes the weights of the stitching layer via gradient descent, using ground-truth labels or the output of f' as soft labels [7, 30, 80]. Additional tweaks such as normalization or regularization may be beneficial in certain contexts [7, 30]. For simple linear stitching layers T , the weights can be computed by solving the least squares problem $\|R^{(l)}T - R^{(l')}\|_F$.

5 Properties and Application of Similarity Measures

In this section, we discuss practical aspects regarding the application of similarity measures. We begin by outlining the current state of research that analyzes properties of existing measures and their relationship. Afterwards, we summarize applications in existing literature and then discuss the choice of measures in more detail, before providing additional considerations for comparing neural networks.

Properties and Evaluation of Similarity Measures. Most research on properties of similarity measures in the deep learning literature focuses on representational similarity. We give a detailed review of existing analyses of representational similarity measures in Appendix C. We further provide an overview of existing comparative tests of these measures in Table 3, which highlights that except for the recent ReSi benchmark [70], most analyses only considered very limited sets of measures. By contrast, functional similarity measures have been broadly analyzed in various contexts, including inter-rater agreement [89, 129, 135], model fingerprinting [138], and ensemble learning [76].

Resources. There are only few resources that enable easy use of similarity measures. Most notably, the recent ReSi benchmark¹ [70] provides implementations of 24 representational similarity measures, and also allows for testing of new measures on representations from a broad range of neural network models and datasets, spanning the graph, language, and vision domains. Similarly, Ding et al. [34] provide code² to replicate their experiments and test new measures, albeit being smaller in scope. Finally, Cloos et al. [26] have collected implementations of similarity measures

¹<https://github.com/mklabunde/resi>

²https://github.com/js-d/sim_metric

Table 3. Ranked Performance of Representational Similarity Measures in Existing Tests

| Test | Correlation (C.1) | | | | | | | | | | Discriminative Abilities (C.2) | | | | | | | | | |
|--|-------------------|------|------|------|--------------|------|------|------|--------------------|------|--------------------------------|------|----------------|-------|---------------------|-------|---------------|------|-------------------------|-------|
| | Accuracy | | | | Disagreement | | JSD | | Squared Difference | | Noise Addition | | Layer Matching | | Dimension Subsample | | Cluster Count | | Architecture Clustering | |
| Reference | [17] | [34] | [57] | [70] | [8] | [70] | [70] | [70] | [17] | [17] | [101] | [24] | [73] | [116] | [125] | [125] | [8] | [17] | [116] | [116] |
| Mean Canonical Correlation | 4 | | 4 | | | | | | 5 | | 3 | | 5 | | | | | | | |
| Mean Canonical Correlation ² | | | 2 | | | | | | | | | | 3 | | | | | | | |
| Singular Vector Canonical Correlation Analysis (SVCCA) | | | | 19 | | 16 | 14 | | | | 2 | | 6 | | | | 4 | | | |
| Projection-Weighted Canonical Correlation Analysis (PWCCA) | 4 | 3 | 3 | 22 | | 14 | 17 | 4 | | | 1 | | 2 | 3 | | | | | 4 | |
| Orthogonal Procrustes [CC] | | | | 5 | | 9 | 8 | | | | | | | | | | | | 21 | 23 |
| Orthogonal Procrustes [CC, MN] | | | | 3 | | 5 | 3 | 2 | | | | | | | | | | | 8 | 8 |
| Permutation Procrustes | | | | 8 | | 11 | 15 | | | | | | | | | | | | 4 | 10 |
| Angular Shape Metric | | | | 3 | | 4 | 2 | | | | | | | | | | | | 3 | 10 |
| Linear Regression | | | | 14 | | 12 | 11 | | | | | | 6 | | | | | | 12 | 9 |
| Aligned Cosine Similarity | | | | 12 | | 7 | 6 | | | | | | | | | | | | 6 | 6 |
| Correlation Match [Relaxed] | | | | 1 | | 10 | 13 | | | | | | | | | | | | 13 | 18 |
| Correlation Match [Strict] | | | | 2 | | 14 | 16 | | | | | | | | | | | | 15 | 21 |
| ContraSim | | | | | | | | | | | | | | 1 | | | | | 1 | 1 |
| Norm of Representational Similarity Matrix Difference | | | | 11 | | 21 | 21 | | | | | | | | | | | | 18 | 13 |
| Representational Similarity Analysis (RSA) | | | | 9 | | 2 | 10 | | | | | | | | | | | | 10 | 12 |
| Centered Kernel Alignment (CKA) [Linear] | | | | | | | | | | | | | | | | | | | 12 | 17 |
| Centered Kernel Alignment (CKA) [RBF 0.8] | 3 | 2 | | 7 | 2 | 6 | 5 | 2 | | | 2 | 4 | 2 | 2 | 2 | 3 | 2 | 2 | 9 | 7 |
| Distance Correlation (dCor) | | | | 6 | | 1 | 3 | | | | | | | | | | | | 7 | 2 |
| Eigenspace Overlap Score (EOS) | | | | 15 | | 20 | 18 | | | | | | | | | | | | 15 | 15 |
| Unified Linear Probing (GULP) [$\lambda = 0$] | 6 | | | 23 | | 18 | 12 | 6 | | | | | | | | | | | 14 | 14 |
| Unified Linear Probing (GULP) [tuned λ] | 1 | | | | | | | 1 | | | | | | | | | | | 14 | 21 |
| Riemannian Distance | | | | | | | | | | | | | | | | | | | | |
| Jaccard | | | | 21 | | 3 | 1 | | | | | | | | | | | | 2 | 1 |
| Second-Order Cosine Similarity | | | | 20 | | 8 | 9 | | | | 1 | | | | | | | | 1 | 3 |
| Rank Similarity | | | | 10 | | 13 | 7 | | | | | | | | | | | | 5 | 4 |
| Multi-Scale Intrinsic Distance (IMD) | | | | 17 | | 19 | 20 | | | | | | | | | | | | 20 | 16 |
| Representation Topology Divergence (RTD) | | | | 13 | 1 | 17 | 19 | | | | | | | | | | | | 11 | 5 |
| Magnitude Difference | | | | 17 | | 24 | 23 | | | | | | | | | | | | 24 | 19 |
| Concentricity Difference | | | | 16 | | 23 | 22 | | | | | | | | | | | | 23 | 17 |
| Uniformity Difference | | | | 24 | | 22 | 23 | | | | | | | | | | | | 22 | 22 |

Rank 1 indicates best performance. Empty cells indicate that a measure was not considered in the corresponding test. For the orthogonal Procrustes measure, different normalization strategies were used—either centering (CC) or both centering and normalization to unit norm (MN). Measures are ranked by their average performance across all variations of a single test; variance and quantitative differences in performance are not shown. More details on the tests and rank aggregation are given in Appendix D. Overall, it can be seen that most tests considered only a few measures, indicating a gap in existing research. Further, no measure generally stands out.

in an online repository,³ aiming to provide a standardized interface for application of existing measures.

Applications in the Literature. Similarity measures have been used in a wide array of contexts with two main objectives: to *understand* aspects of deep learning and to *improve* deep learning systems. In this section, we give an overview and examples of such applications; for a wider overview we refer to Sucholutsky et al. [136, Section 4].

Focal points of work aiming at *understanding* deep learning include the effects of model architecture and objective function on what neural networks learn, as well as studies on model universality, i.e., the extent models converge to similar behavior under different training setups. For example, Raghu et al. [115] and Park and Kim [111] studied the differences between vision transformers and CNNs by comparing their representations and analyzing changes in classification performance after modifying the architectures. Similarly, the effect of width and depth [107] and the importance of specific layers [133] has been investigated. The impact of differences in the objective functions has, for instance, been studied by Kornblith et al. [72], who analyzed layerwise differences in representations between models that vary only in their loss function and further evaluated transferability

³<https://github.com/nacloos/similarity-repository>

of these models via functional similarity measures. Grigg et al. [51] took a similar approach when comparing supervised to self-supervised models. Some studies have also investigated the impact of training adversarially robust models, by considering how intra- [25] and inter-architecture [64] similarities of representations from robust and non-robust models differ, or how stitching these kinds of models affects performance [5]. Finally, studies on model universality have found that neural networks trained under different training setups are often at least partially similar in their representations [97–99, 137, 156], even if stemming from different modalities [92]. This is, however, contrasted by substantial differences in functional similarity that result from varying training seeds [14, 39, 69, 86, 137] or the training data by a single instance [15].

Yet, there are other directions, including analyses on the impact of input features [58] and fine-tuning [100] or studies comparing representations of visual information from CNNs to those from mice [128] and human brains [157].

Work that used similarity measures to *improve* systems is comparatively rarer but includes studies on optimizing ensembles and knowledge distillation, i.e., the problem of transferring knowledge of a typically large teacher model into a smaller student model. Works on improving ensembles have applied similarity measures when encouraging representational diversity of models [35, 148, 161] or penalizing similarity of soft predictions of ensemble parts [163]. Similarly, in knowledge distillation, the student models have been trained by maximizing similarity with the teacher model in its representations [112, 121, 164] as well as in functional outputs [49, 165].

Similarity Measure Selection. Using both representational and functional similarity measures allows for assessing similarity of neural networks in a holistic manner. While deciding suitability of a measure requires a case-by-case evaluation, we can make some high-level recommendations.

For functional similarity measures, there are generally no measures that are fundamentally incorrect for a given application. Table 2 provides all information necessary to narrow down the most suitable measures. Using multiple measures can give nuanced insights. Most prediction- and performance-based measures, except Rashomon capacity, can be computed in linear time, keeping computational costs low when using several measures from these categories. For a robust analysis, we recommend using measures that control for confounding factors such as random agreement and error rate. If white-box access to the models is available and computational constraints allow for it, then one could further consider more granular gradient-based measures or stitching.

In contrast, selecting appropriate measures for representational similarity is challenging due to the opacity of neural representations. It is often unclear which representations can be considered equivalent and what kinds of differences in models or representations measures are sensitive to. Despite the big number of existing measures, research evaluating their applicability with respect to these aspects is surprisingly limited. Therefore, we can only give a few general recommendations. First, if it is known which groups of transformations the given representations are equivalent under, then measures should be filtered accordingly (see Table 1). Second, one can check if some of the existing analyses referenced in Table 3 are relevant for the given scenario to narrow down the number of measures. Third, some insights may come from the objective functions of the models to be compared. For instance, if similarity of instance representations is modeled in terms of angles, then similarity measures based on Euclidean distance may not be suitable.

Further, one can consider advantages and disadvantages of different categories of measures. For instance, alignment-based measures are less flexible in their invariances than RSM- and neighborhood-based measures, which can easily be adapted in their inner similarity functions. Topology-based measures, which also compute pairwise distance matrices in addition to estimate persistence intervals of topological features, face similar computational challenges, likely making them unsuitable for a large number of comparisons. Hence, if the number of inputs is large, then other categories of measures may be preferred—though for individual measures, faster variants

such as a batched computation of CKA were proposed [108]. Another key difference between representational similarity measures is their flexibility in weighting local versus global similarity of representations. Neighborhood-based measures are easiest to adjust based on the number of nearest neighbors considered. RSM-based measures can generally also address this issue by creating RSMs based on similarity functions that take distance between instances into account, e.g., the RBF kernel. However, measures from other categories generally lack this flexibility. Finally, the simple and interpretable nature of many descriptive statistics and neighborhood-based measures may be of interest in some applications.

Additional Practical Considerations. Apart from measure selection, a few other aspects need to be considered when comparing neural networks. First, as discussed in Appendix C.3, the input data influences the similarity estimates. When generalizability of results is desired, input data needs to be diverse [136]. A larger number of inputs will, however, increase computational costs. When focussing on representational similarity, the choice of layers for comparison yields a similar tradeoff. While pairwise comparisons of all layers provide the most detailed results, limiting comparisons to selected layers, such as the penultimate layer [64, 68, 104], may balance cost and detail effectively.

Finally, as discussed in Section 2.1, preprocessing of representations might be necessary to meet the requirements of some measures. Understanding of the given representation space can, however, inform additional preprocessing. For example, language model representations like those from BERT [33] often have a few dimensions with high mean and variance compared to the remaining dimensions [142], skewing measures like cosine similarity. Timkey and van Schijndel [142] addressed this by standardizing the representations to zero mean and unit variance, thereby improving the alignment of cosine similarity between words with human similarity judgements. Therefore, this normalization may be advisable when comparing such language representations via similarity measures that use cosine similarity, e.g., for RSMs or nearest neighbors. This example also illustrates how normalization can extend the invariances of a given measure, which, in this case, would otherwise not be invariant to translation and anisotropic scaling. However, normalizing representations without such insights can be counterproductive. As can be seen in Table 3, the performance of orthogonal Procrustes is strongly affected by differences in normalization.

6 Discussion and Open Research Challenges

In this survey, we describe more than 50 similarity measures. This yields a stark contrast to the rather small amount of research dedicated to systematically analyzing and comparing the existing measures that is highlighted in Appendix C. In particular, representational similarity measures pose many open questions of high practical relevance. We argue that this constitutes a significant gap in research, as deeper understanding of the properties of measures is crucial to properly measure similarity and correctly interpret their scores.

In this section, we discuss challenges in the application of similarity measures, and connect these to open research questions that we argue require more attention in the future. A discussion of notions of similarity and corresponding measures beyond the scope of this survey is provided in Appendix E.

Applicability of Representational Similarity Measures. Applying different representational similarity measures to the same pair of models can yield materially different results [68]. Given this potential for disagreement, it is crucial to have an understanding which measures are able to capture those differences in representations that are relevant for a given application scenario. As discussed in Section 5, there is, however, only very limited research that has investigated the applicability of representational similarity measures in a broad manner. Further, there is also only limited research aimed at understanding the geometry of neural representations, which could

additionally inform about the compatibility of similarity measures with specific representations, or preprocessing approaches that could be utilized to create such compatibility. The recent ReSi benchmark [70], which builds on this survey, can be seen as a first effort toward enabling such systematic analyses, and we believe additional research in this direction is required to enable more informed decisions when choosing representational similarity measures.

Interpretability. Unless a similarity score indicates perfect (dis-)similarity through a bounded minimum or maximum value, one typically cannot directly infer an intuitively interpretable degree of similarity from the score itself. One reason for this is that, due to non-linearities in a measure, the resulting scores may be misleading. For example, the widely used cosine similarity (Equation (A.5)) changes non-linearly with the angle between two compared vectors, to the degree that a seemingly high similarity of 0.95 still corresponds to an 18° angle. Another issue is that the similarity scores that one can obtain may strongly depend on the context. For instance, a prediction disagreement (Equation (51)) of 0.05 can be considered low in a difficult classification problem with many classes and high in an easy binary classification problem where one expects near-perfect accuracy. Such contextualization may be easy to establish for measures as intuitive as disagreement, however, for more opaque measures, interpretation is typically much more difficult. Generally, properties of the inputs can influence the obtained similarity scores (see Appendix C.3), and further, factors such as dimensionality of representations might also affect the range that a similarity measure can produce. The latter issue is exemplified in Appendix F, where we show that the orthogonal Procrustes scores of two random representations increase with increasing dimension, which proves how dissimilar representations may receive different scores based on such underlying factors. Therefore, we argue that more research is required to improve the interpretability of measures, e.g., via expected values or boundaries of similarity scores in terms of input similarity or dimensionality.

Robustness of Representational Similarity Measures. Specifically for the CKA (Equation (28)), it has been shown that perturbations of single instance representations can strongly affect the resulting similarity scores (see Appendix C.1). Such sensitivities can be particularly harmful in applications where reliability of similarity measures is a prerequisite. For instance, similarity measures could be used to identify model reuse in the legal context of intellectual property protection [138]. Therefore, we argue that more research on the robustness of similarity measures is required to understand and improve their reliability.

7 Conclusion

Representational similarity and functional similarity represent two complementing perspectives on analyzing and comparing neural networks. In this work, we provide a comprehensive overview of existing measures for both representational and functional similarity. We provide formal definitions for 53 similarity measures, along with a systematic categorization into different types of measures and pedagogical illustrations.

In addition, we survey the literature to shed light on some of their salient properties and provide guidance for the practical application of similarity measures. We specifically identify a lack of research that analyzes properties and applicability of representational similarity measures for specific contexts in a unified manner. This gap in the literature also affects the quality of the recommendations that one can make about their practical applicability. We argue that additional research is necessary to enable the informed application of similarity measures and better understand similarity of neural network models. Moreover, assessing similarity of neural networks is an important aspect in several deep learning-related problems, including knowledge distillation, pruning, model updating, continual learning, model merging, and contrastive learning. Despite this importance, only limited consideration has been put into the choice of measures within such applications, which may also be due to a lack of awareness about the available measures and their

properties. In that sense, we hope that our work lays a foundation for more systematic research on the properties of similarity measures and their applicability across deep learning. Further, with our categorization and analysis, we believe that our work can assist researchers and practitioners in choosing appropriate similarity measures.

References

- [1] Chirag Agarwal, Owen Queen, Himabindu Lakkaraju, and Marinka Zitnik. 2023. Evaluating explainability for graph neural networks. *Sci. Data* 10, 1 (2023).
- [2] Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. In *NeurIPS*.
- [3] Suguru Arimoto. 1972. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Trans. Intell. Transport. Syst.* 18, 1 (1972).
- [4] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? *NeurIPS* 27 (2014).
- [5] András Balogh and Márk Jelasity. 2023. On the functional similarity of robust and non-robust neural representations. In *ICML*.
- [6] Mousumi Banerjee, Michelle Capozzoli, Laura McSweeney, and Debajyoti Sinha. 1999. Beyond kappa: A review of interrater agreement measures. *Can. J. Stat.* 27, 1 (1999).
- [7] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. 2021. Revisiting model stitching to compare neural representations. In *NeurIPS*.
- [8] Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. 2022. Representation topology divergence: A method for comparing neural network representations. In *ICML*.
- [9] Lorenzo Basile, Santiago Acevedo, Luca Bortolussi, Fabio Anselmi, and Alex Rodriguez. 2025. Intrinsic dimension correlation: uncovering nonlinear connections in multimodal representations. In *The Thirteenth International Conference on Learning Representations*. Retrieved from <https://openreview.net/forum?id=Qj1KwBZaEI>
- [10] Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *ICLR*.
- [11] Robert Bennett. 1969. The intrinsic dimensionality of signal collections. *IEEE Trans. Intell. Transport. Syst.* 15, 5 (1969).
- [12] Rajendra Bhatia. 2007. *Positive Definite Matrices*.
- [13] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. 2019. On the Bures-Wasserstein distance between positive definite matrices. *Exposit. Math.* 37, 2 (2019).
- [14] Srinadh Bhojanapalli, Kimberly Wilber, Andreas Veit, Ankit Singh Rawat, Seungyeon Kim, Aditya Menon, and Sanjiv Kumar. 2021. On the reproducibility of neural network predictions. arXiv:2102.03349. Retrieved from <https://arxiv.org/abs/2102.03349>
- [15] Emily Black and Matt Fredrikson. 2021. Leave-one-out unfairness. In *FAccT '21*.
- [16] Richard Blahut. 1972. Computation of channel capacity and rate-distortion functions. *IEEE Trans. Intell. Transport. Syst.* 18, 4 (1972).
- [17] Enric Boix-Adsera, Hannah Lawrence, George Stepaniants, and Philippe Rigollet. 2022. GULP: A prediction-based metric between representations. In *NeurIPS*.
- [18] Ali Borji. 2019. Pros and cons of GAN evaluation measures. *Comput. Vis. Image Understand.* 179 (2019).
- [19] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. 2005. Diversity creation methods: A survey and categorisation. *Inf. Fus.* 6, 1 (2005).
- [20] Donald Bures. 1969. An extension of Kakutani's theorem on infinite product measures to the tensor product of semifinite w^* -algebras. *Trans. Am. Math. Soc.* 135 (1969).
- [21] Francesco Camastra and Antonino Staiano. 2016. Intrinsic dimension estimation: Advances and open problems. *Inf. Sci.* 328 (2016).
- [22] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. arXiv:2006.14799 (2020).
- [23] Sung-Hyuk Cha. 2007. Comprehensive survey on distance/similarity measures between probability density functions. *Int. J. Math. Models Methods Appl. Sci.* 1, 4 (2007).
- [24] Zuohui Chen, Yao Lu, Wen Yang, Qi Xuan, and Xiaoni Yang. 2021. Graph-based similarity of neural network representations. arXiv:2111.11165. Retrieved from <https://arxiv.org/abs/2111.11165>
- [25] Christian Cianfarani, Arjun Nitin Bhagoji, Vikash Sehwal, Ben Zhao, Heather Zheng, and Prateek Mittal. 2022. Understanding robust learning through the lens of representation similarities. In *NeurIPS*.
- [26] Nathan Cloos, Guangyu Robert Yang, and Christopher J. Cueva. 2024. A framework for standardizing similarity measures in a rapidly evolving field. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*.

- [27] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* 20, 1 (1960).
- [28] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. 2012. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.* 13, 1 (2012).
- [29] Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz S. Kandola. 2001. On kernel-target alignment. In *NeurIPS*.
- [30] Adrián Csizsárik, Péter Korösi-Szabó, Ákos K. Matszangosz, Gergely Papp, and Dániel Varga. 2021. Similarity and matching of neural network representations. In *NeurIPS*.
- [31] Arghya Datta, Subhrangshu Nandi, Jingcheng Xu, Greg Ver Steeg, He Xie, Anoop Kumar, and Aram Galstyan. 2023. Measuring and mitigating local instability in deep neural networks. In *ACL*.
- [32] Jesús A. De Loera and Edward D. Kim. 2013. Combinatorics and geometry of transportation polytopes: An update. *Discr. Geom. Algebr. Combin.* 625 (2013), 37–76.
- [33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
- [34] Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. 2021. Grounding representation similarity through statistical testing. In *NeurIPS*.
- [35] Giang Do, Hung Le, and Truyen Tran. 2024. SimSMoE: Solving representational collapse via similarity measure. arXiv:2406.15883. Retrieved from <https://arxiv.org/abs/2406.15883>
- [36] Yupei Du and Dong Nguyen. 2023. Measuring the instability of fine-tuning. In *ACL*.
- [37] Lyndon Duong, Jingyang Zhou, Josue Nassar, Jules Berman, Jeroen Olieslagers, and Alex H. Williams. 2023. Representational dissimilarity metric spaces for stochastic neural networks. In *ICLR*.
- [38] Ehsan Elhamifar and René Vidal. 2013. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 11 (2013), 2765–2781.
- [39] Mahdi Milani Fard, Quentin Cormier, Kevin Robert Canini, and Maya R. Gupta. 2016. Launch and iterate: Reducing prediction churn. In *NeurIPS*.
- [40] Guangchao Charles Feng. 2014. Mistakes and how to avoid mistakes in using intercoder reliability indices. *Methodol. Eur. J. Res. Methods Behav. Soc. Sci.* 11, 1 (2014).
- [41] Yunzhen Feng, Runtian Zhai, Di He, Liwei Wang, and Bin Dong. 2020. Transferred discrepancy: Quantifying the difference between representations. arXiv:2007.12446. Retrieved from <https://arxiv.org/abs/2007.12446>
- [42] Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychol. Bull.* 76, 5 (1971).
- [43] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep ensembles: A loss landscape perspective. arXiv:1912.02757. Retrieved from <https://arxiv.org/abs/1912.02757>
- [44] Robert Geirhos, Kristof Meding, and Felix A. Wichmann. 2020. Beyond accuracy: Quantifying trial-by-trial behaviour of CNNs and humans by measuring error consistency. In *NeurIPS*.
- [45] Michelle Girvan and Mark E. J. Newman. 2002. Community structure in social and biological networks. *Natl. Acad. Sci.* 99, 12 (2002).
- [46] Natasa Gisev, J. Simon Bell, and Timothy F. Chen. 2013. Interrater agreement and interrater reliability: Key concepts, approaches, and applications. *Res. Soc. Admin. Pharm.* 9, 3 (2013).
- [47] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. 2022. On the symmetries of deep learning models and their internal representations. In *NeurIPS*.
- [48] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*.
- [49] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *Int. J. Comput. Vis.* 129, 6 (2021), 1789–1819.
- [50] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. 2005. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*.
- [51] Tom George Grigg, Dan Busbridge, Jason Ramapuram, and Russ Webb. 2021. Do self-supervised and supervised methods learn similar visual representations? In *NeurIPS 2021 Workshop on Self-Supervised Learning*.
- [52] Matthew Gwilliam and Abhinav Shrivastava. 2022. Beyond supervised vs. unsupervised: Representative benchmarking and analysis of image representation learning. In *CVPR*.
- [53] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Cultural shift or linguistic drift? Comparing two computational measures of semantic change. In *EMNLP*.
- [54] William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *ACL*.
- [55] Sarah E. Harvey, Brett W. Larsen, and Alex H. Williams. 2024. Duality of bures and shape distances with implications for comparing neural representations. In *Proceedings of UniReps: The First Workshop on Unifying Representations in Neural Models*.
- [56] Allen Hatcher. 2005. *Algebraic Topology*.

- [57] Lucas Hayne, Heejung Jung, and R. Carter. 2024. Does representation similarity capture function similarity? *Transactions on Machine Learning Research* (2024). Retrieved from <https://openreview.net/forum?id=YY2iA0hfa>
- [58] Katherine Hermann and Andrew Lampinen. 2020. What shapes feature representations? Exploring datasets, architectures, and training. In *NeurIPS*.
- [59] Harald Hotelling. 1935. The most predictable criterion. *J. Educ. Psychol.* 26, 2 (1935).
- [60] Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika* 28, 3/4 (1936).
- [61] Andrew Hryniowski and Alexander Wong. 2020. Inter-layer information similarity assessment of deep neural networks via topological similarity and persistence analysis of data neighbour dynamics. arXiv:2012.03793. Retrieved from <https://arxiv.org/abs/2012.03793>
- [62] Hsiang Hsu and Flavio Calmon. 2022. Rashomon capacity: A metric for predictive multiplicity in classification. In *NeurIPS*.
- [63] Jaehui Hwang, Dongyoon Han, Byeongho Heo, Song Park, Sanghyuk Chun, and Jong-Seok Lee. 2023. Similarity of neural architectures based on input gradient transferability. arXiv:2210.11407. Retrieved from <https://arxiv.org/abs/2210.11407>
- [64] Haydn T. Jones, Jacob M. Springer, Garrett T. Kenyon, and Juston S. Moore. 2022. If you’ve trained one you’ve trained them all: Inter-architecture similarity increases with robustness. In *UAI*.
- [65] Meenakshi Khosla and Alex H. Williams. 2024. Soft matching distance: A metric on neural representations that captures single-neuron tuning. In *Proceedings of UniReps: The First Workshop on Unifying Representations in Neural Models*.
- [66] Valentin Khrulkov and Ivan Oseledets. 2018. Geometry score: A method for comparing generative adversarial networks. In *ICML*.
- [67] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *ICLR*.
- [68] Max Klabunde, Mehdi Ben Amor, Michael Granitzer, and Florian Lemmerich. 2023. Towards measuring representational similarity of large language models. In *UniReps: The First Workshop on Unifying Representations in Neural Models*.
- [69] Max Klabunde and Florian Lemmerich. 2022. On the prediction instability of graph neural networks. In *ECML PKDD*.
- [70] Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. 2025. ReSi: A comprehensive benchmark for representational similarity measures. In *The Thirteenth International Conference on Learning Representations*. Retrieved from <https://openreview.net/forum?id=PRvdO3nffI>
- [71] Camila Kolling, Till Speicher, Vedant Nanda, Mariya Toneva, and Krishna P. Gummadi. 2023. Pointwise representational similarity. arXiv:2305.19294. Retrieved from <https://arxiv.org/abs/2305.19294>
- [72] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. 2021. Why do better loss functions lead to less transferable features? In *NeurIPS*.
- [73] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of neural network representations revisited. In *ICML*.
- [74] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2 (2008).
- [75] Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *EMNLP*.
- [76] Ludmila I. Kuncheva and Christopher J. Whitaker. 2003. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.* 51 (2003).
- [77] Richard D. Lange, David Rolnick, and Konrad Kording. 2022. Clustering units in neural networks: Upstream vs downstream information. *Transactions on Machine Learning Research* (2022). Retrieved from <https://openreview.net/forum?id=Euf7KofunK>
- [78] Derrik N. Lawley. 1938. A generalization of fisher’s z test. *Biometrika* 30, 1/2 (1938).
- [79] Yoonho Lee, Huaxiu Yao, and Chelsea Finn. 2023. Diversify and disambiguate: Out-of-distribution robustness via disagreement. In *ICLR*.
- [80] Karel Lenc and Andrea Vedaldi. 2015. Understanding image representations by measuring their equivariance and equivalence. In *CVPR*.
- [81] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E. Hopcroft. 2016. Convergent learning: Do different neural networks learn the same representations? In *ICLR*.
- [82] Yuanchun Li, Ziqi Zhang, Bingyan Liu, Ziyue Yang, and Yunxin Liu. 2021. ModelDiff: Testing-based DNN similarity comparison for model reuse detection. In *ISSTA*.
- [83] Baihan Lin. 2022. Geometric and topological inference for deep representations of complex networks. In *WWW’22. Association for Computing Machinery, New York, NY*, 334–338. <https://doi.org/10.1145/3487553.3524194>
- [84] Baihan Lin and Nikolaus Kriegeskorte. 2018. Adaptive geo-topological independence criterion. arXiv:1810.02923. Retrieved from <https://arxiv.org/abs/1810.02923>

- [85] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* 37, 1 (1991).
- [86] Huiting Liu, Avinesh P. V. S., Siddharth Patwardhan, Peter Grasch, and Sachin Agarwal. 2022. Model stability with continuous data updates. arXiv:2201.05692. Retrieved from <https://arxiv.org/abs/2201.05692>
- [87] Yao Lu, Wen Yang, Yunzhe Zhang, Zuohui Chen, Jinyin Chen, Qi Xuan, Zhen Wang, and Xiaoniu Yang. 2022. Understanding the dynamics of DNNs using graph modularity. In *ECCV*.
- [88] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. In *ICML*.
- [89] Malcom MacLure and Walter C. Willett. 1987. Misinterpretation and misuse of the kappa statistic. *Am. J. Epidemiol.* 126, 2 (1987).
- [90] Omid Madani, David M. Pennock, and Gary William Flake. 2004. Co-Validation: Using model disagreement on unlabeled data to validate classification algorithms. In *NeurIPS*.
- [91] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *ICLR*.
- [92] Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E. O'Connor. 2024. Do vision and language encoders represent the world similarly? In *CVPR*.
- [93] Charles T. Marx, Flávio P. Calmon, and Berk Ustun. 2020. Predictive multiplicity in classification. In *ICML*.
- [94] Binny Mathew, Sandipan Sikdar, Florian Lemmerich, and Markus Strohmaier. 2020. The POLAR framework: Polar opposites enable interpretability of pre-trained word embeddings. In *WWW*.
- [95] Avner May, Jian Zhang, Tri Dao, and Christopher Ré. 2019. On the downstream performance of compressed word embeddings. In *NeurIPS*.
- [96] R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.
- [97] David McNeely-White, Benjamin Sattler, Nathaniel Blanchard, and Ross Beveridge. 2020. Exploring the interchangeability of CNN embedding spaces. arXiv:2010.02323. Retrieved from <https://arxiv.org/abs/2010.02323>
- [98] Johannes Mehrer, Nikolaus Kriegeskorte, and Tim C. Kietzmann. 2018. Beware of the beginnings: Intermediate and higher-level representations in deep neural networks are strongly affected by weight initialization. In *Conference on Cognitive Computational Neuroscience*.
- [99] Johannes Mehrer, Courtney J. Spoerer, Nikolaus Kriegeskorte, and Tim C. Kietzmann. 2020. Individual differences among deep neural network models. *Nat. Commun.* 11 (2020).
- [100] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.
- [101] Ari S. Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *NeurIPS*.
- [102] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *ICLR*.
- [103] Alex Graeme Murphy, Joel Zylberberg, and Alona Fyshe. 2024. Correcting biased centered kernel alignment measures in biological and artificial neural networks. In *ICLR 2024 Workshop on Representational Alignment*. <https://openreview.net/forum?id=E1NRrGtIHG>
- [104] Vedant Nanda, Till Speicher, Camila Kolling, John P. Dickerson, Krishna P. Gummadi, and Adrian Weller. 2022. Measuring representational robustness of neural networks through shared invariances. In *ICML*.
- [105] Mark E. J. Newman. 2006. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. U.S.A.* 103, 23 (2006).
- [106] Mark E. J. Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69 (2004).
- [107] Thao Nguyen, Maithra Raghu, and Simon Kornblith. 2021. Do wide and deep networks learn the same things? Uncovering how neural network representations vary with width and depth. In *ICLR*.
- [108] Thao Nguyen, Maithra Raghu, and Simon Kornblith. 2022. On the origins of the block structure phenomenon in neural network representations. *Trans. Mach. Learn. Res.* (2022).
- [109] Mitchell Ostrow, Adam Joseph Eisen, Leo Kozachkov, and Ila R. Fiete. 2023. Beyond geometry: Comparing the temporal structure of computation in neural circuits with dynamical similarity analysis. In *NeurIPS*.
- [110] Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. 2023. Agree to disagree: Diversity through disagreement for better transferability. In *ICLR*.
- [111] Namuk Park and Songkuk Kim. 2022. How do vision transformers work? In *ICLR*.
- [112] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *CVPR*.

- [113] K. C. Sreedharan Pillai. 1955. Some new test criteria in multivariate analysis. *Ann. Math. Stat.* 26, 1 (1955).
- [114] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *NeurIPS*.
- [115] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. 2021. Do vision transformers see like convolutional neural networks? *NeurIPS* 34 (2021), 12116–12128.
- [116] Adir Rahamim and Yonatan Belinkov. 2024. ContraSim—Analyzing neural representations based on contrastive learning. In *NAACL*.
- [117] James O. Ramsay, Jos ten Berge, and George P. H. Styan. 1984. Matrix correlation. *Psychometrika* 49 (1984).
- [118] Annika Reinke, Minu D. Tizabi, Michael Baumgartner, Matthias Eisenmann, Doreen Heckmann-Nötzel, A. Emre Kavur, Tim Rädtsch, Carole H. Sudre, Laura Acion, Michela Antonelli, et al. 2024. Understanding metric-related pitfalls in image analysis validation. *Nature Methods* 21, 2 (2024), 182–194.
- [119] Paul Robert and Yves Escoufier. 1976. A unifying tool for linear multivariate statistical methods: The RV-coefficient. *J. Roy. Stat. Soc. Ser. C (Appl. Stat.)* 25, 3 (1976).
- [120] Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. In *SaTML '23*. 464–483. <https://doi.org/10.1109/SaTML54575.2023.00039>
- [121] Aninda Saha, Alina N. Bialkowski, and Sara Khalifa. 2022. Distilling representational similarity using centered kernel alignment (CKA). In *BMVC'22*. BMVA Press.
- [122] Uday Singh Saini, Pravallika Devineni, and Evangelos E. Papalexakis. 2021. Subspace clustering based analysis of neural networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track*.
- [123] Tobias Schumacher, Hinrikus Wolf, Martin Ritzert, Florian Lemmerich, Martin Grohe, and Markus Strohmaier. 2021. The effects of randomness on the stability of node embeddings. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.
- [124] Peter H. Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika* 31 (1966).
- [125] Mahdiyar Shahbazi, Ali Shirali, Hamid Aghajan, and Hamed Nili. 2021. Using distance on the Riemannian manifold to compare representations in brain and in models. *NeuroImage* 239 (2021).
- [126] Gil I. Shamir and Lorenzo Coviello. 2020. Anti-Distillation: Improving reproducibility of deep networks. arXiv:2010.09923. Retrieved from <https://arxiv.org/abs/2010.09923>
- [127] Roger N. Shepard. 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika* 27 (1962).
- [128] Jianghong Shi, Eric Shea-Brown, and Michael Buice. 2019. Comparison against task driven artificial neural networks reveals functional properties in mouse visual cortex. In *NeurIPS*.
- [129] Julius Sim and Chris C. Wright. 2005. The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. *Phys. Therapy* 85, 3 (2005).
- [130] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at ICLR*.
- [131] David B. Skalak et al. 1996. The sources of increased accuracy for two proposed boosting algorithms. In *AAAI Integrating Multiple Learned Models Workshop*.
- [132] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. 2012. Feature selection via dependence maximization. *J. Mach. Learn. Res.* 13, 47 (2012), 1393–1434. <http://jmlr.org/papers/v13/song12a.html>
- [133] Sharath Nittur Sridhar and Anthony Sarah. 2020. Undivided attention: Are intermediate layers necessary for BERT? arXiv:2012.11881. Retrieved from <https://arxiv.org/abs/2012.11881>
- [134] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. 2021. Does knowledge distillation really work? In *NeurIPS*.
- [135] Steven E. Stemler. 2019. A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Pract. Assess. Res. Eval.* 9 (2019).
- [136] Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C. Love, Erin Grant, Jascha Achterberg, Joshua B. Tenenbaum, et al. 2023. Getting aligned on representational alignment. arXiv:2310.13018. Retrieved from <https://arxiv.org/abs/2310.13018>
- [137] Cecilia Summers and Michael J. Dinneen. 2021. Nondeterminism and instability in neural network optimization. In *ICML*.
- [138] Yuchen Sun, Tianpeng Liu, Panhe Hu, Qing Liao, Shouling Ji, Nenghai Yu, Deke Guo, and Li Liu. 2023. Deep intellectual property: A survey. arXiv:2304.14613.
- [139] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. 2007. Measuring and testing dependence by correlation of distances. *Ann. Stat.* 35, 6 (2007).
- [140] Ke Tang, Ponnuthurai N. Suganthan, and Xin Yao. 2006. An analysis of diversity measures. *Mach. Learn.* 65 (2006).

- [141] Shuai Tang, Wesley J. Maddox, Charlie Dickens, Tom Diethe, and Andreas Damianou. 2020. Similarity of neural networks with gradients. arXiv:2003.11498. Retrieved from <https://arxiv.org/abs/2003.11498>
- [142] William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *EMNLP'21*. 4527–4546.
- [143] Howard E. Tinsley and David J. Weiss. 1975. Interrater reliability and agreement of subjective judgments. *J. Counsel. Psychol.* 22, 4 (1975).
- [144] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Alex Bronstein, Ivan Oseledets, and Emmanuel Mueller. 2020. The shape of data: Intrinsic distance for data distributions. In *ICLR*.
- [145] Shashanka Ubaru, Jie Chen, and Yousef Saad. 2017. Fast estimation of $\text{Str}(\mathbf{f}(\mathbf{A}))$ via stochastic Lanczos quadrature. *SIAM J. Matrix Anal. Appl.* 38, 4 (2017).
- [146] Viivi Uurtio, João M. Monteiro, Jaz Kandola, John Shawe-Taylor, Delmiro Fernandez-Reyes, and Juho Rousu. 2017. A tutorial on canonical correlation methods. *Comput. Surv.* 50, 6 (2017).
- [147] Hrishikesh D. Vinod. 1976. Canonical ridge and econometrics of joint production. *J. Econometr.* 4, 2 (1976).
- [148] Tassilo Wald, Constantin Ulrich, Fabian Isensee, David Zimmerer, Gregor Koehler, Michael Baumgartner, and Klaus H Maier-Hein. 2023. Exploring new ways: Enforcing representational dissimilarity to learn new features and reduce error consistency. *ICML Workshop SCIS* (2023).
- [149] Chenxu Wang, Wei Rao, Wenna Guo, Pinghui Wang, Jun Liu, and Xiaohong Guan. 2020. Towards understanding the instability of network embedding. *IEEE Trans. Knowl. Data Eng.* 34, 2 (2020).
- [150] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *CVPR*.
- [151] Guangcong Wang, Guangrun Wang, Wenqi Liang, and Jianhuang Lai. 2022. Understanding weight similarity of neural networks via chain normalization rule and hypothesis-training-testing. arXiv:2208.04369. Retrieved from <https://arxiv.org/abs/2208.04369>
- [152] Liwei Wang, Lunjia Hu, Jiayuan Gu, Zhiqiang Hu, Yue Wu, Kun He, and John E. Hopcroft. 2018. Towards understanding learning representations: To what extent do different neural networks learn the same representation. In *NeurIPS*.
- [153] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*.
- [154] Samuel S. Wilks. 1932. Certain generalizations in the analysis of variance. *Biometrika* 24, 3/4 (1932).
- [155] Alex H. Williams, Erin Kunz, Simon Kornblith, and Scott W. Linderman. 2021. Generalized shape metrics on neural representations. In *NeurIPS*.
- [156] John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *ACL*.
- [157] Yaoda Xu and Maryam Vaziri-Pashkam. 2021. Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nat. Commun.* 12, 1 (2021), 2065.
- [158] Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *ICCL*.
- [159] Haruo Yanai. 1974. Unification of various techniques of multivariate analysis by means of generalized coefficient of determination. *Kodo Keiryogaku (Jpn. J. Behaviormetr.)* 1 (1974).
- [160] Xinghao Yang, Weifeng Liu, Wei Liu, and Dacheng Tao. 2021. A survey on canonical correlation analysis. *IEEE Trans. Knowl. Data Eng.* 33, 6 (2021).
- [161] Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin Rubinstein, Ce Zhang, and Bo Li. 2021. TRS: Transferability reduced ensemble via promoting gradient diversity and model smoothness. In *NeurIPS*.
- [162] Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *NeurIPS*.
- [163] Wentao Zhang, Jiawei Jiang, Yingxia Shao, and Bin Cui. 2020. Efficient diversity-driven ensemble for deep neural networks. In *ICDE*.
- [164] Zikai Zhou, Yunhang Shen, Shitong Shao, Linrui Gong, and Shaohui Lin. 2024. Rethinking centered kernel alignment in knowledge distillation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI'24)*, Jeju, Korea. DOI : <https://doi.org/10.24963/ijcai.2024/628>
- [165] Martin Zong, Zengyu Qiu, Xinzhu Ma, Kunlin Yang, Chunyu Liu, Jun Hou, Shuai Yi, and Wanli Ouyang. 2023. Better teacher better student: Dynamic prior knowledge for knowledge distillation. In *ICLR*.
- [166] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guillaume Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlicek, Agustín Piqueres Lajarán, Vaibhav Srivastav, et al. 2025. SmoLM2: When smol goes big—data-centric training of a small language model. arXiv:2502.02737. Retrieved from <https://arxiv.org/abs/2502.02737>
- [167] Roger Bakeman, Vicenq Quera, Duncan McArthur, and Byron F. Robinson. 1997. Detecting sequential patterns and determining their reliability with fallible observers. *Psychol. Methods* 2, 4 (1997).
- [168] Serguei Barannikov, Ilya Trofimov, Grigori Sotnikov, Ekaterina Trimbach, Alexander Korotin, Alexander Filippov, and Evgeny Burnaev. 2021. Manifold topology divergence: A framework for comparing data manifolds. In *NeurIPS*.

- [169] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*.
- [170] Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Comput. Ling.* 48, 1 (2022).
- [171] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. ACL, 12–58. <https://doi.org/10.3115/v1/W14-3302>
- [172] Davis Brown, Madelyn Ruth Shapiro, Alyson Bittner, Jackson Warley, and Henry Kvinge. 2024. Wild comparisons: A study of how representation similarity changes when input data is drawn from a shifted distribution. In *ICLR'24 Workshop on Representational Alignment*.
- [173] Ted Byrt, Janet Bishop, and John B. Carlin. 1993. Bias, prevalence and kappa. *J. Clin. Epidemiol.* 46, 5 (1993).
- [174] Wei Chen, Zichen Miao, and Qiang Qiu. 2023. Inner product-based neural network similarity. In *NeurIPS*.
- [175] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. Online. <https://doi.org/10.18653/v1/2020.acl-main.747>
- [176] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *EMNLP*. Association for Computational Linguistics, 2475–2485. <https://doi.org/10.18653/v1/D18-1269>
- [177] Tianyu Cui, Yogesh Kumar, Pekka Marttinen, and Samuel Kaski. 2022. Deconfounded representation similarity for comparison of neural networks. In *NeurIPS*.
- [178] MohammadReza Davari, Stefan Horoi, Amine Natick, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. 2022. On the inadequacy of CKA as a measure of similarity in deep learning. In *ICLR'22 Workshop on Geometrical and Topological Representation Learning*.
- [179] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.
- [180] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.
- [181] Marin Dujmović, Jeffrey S. Bowers, Federico Adolfini, and Gaurav Malhotra. 2022. The pitfalls of measuring representational similarity using representational similarity analysis. *bioRxiv* (2022). <https://www.biorxiv.org/content/early/2022/04/07/2022.04.05.487135>
- [182] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. 2022. Vision-language pre-training: Basics, recent advances, and future trends. *Found. Trends Comput. Graph. Vis.* 14, 3–4 (2022), 163–352.
- [183] John C. Gower. 1975. Generalized procrustes analysis. *Psychometrika* 40 (1975).
- [184] Florentin Guth and Brice Ménard. 2024. On the universality of neural encodings in CNNs. In *ICLR'24 Workshop on Representational Alignment*.
- [185] Will Hamilton, Zhitaoying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*.
- [186] J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini. 2023. Visual Object Recognition. <https://doi.org/doi:10.18112/openneuro.ds000105.v3.0.0>
- [187] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
- [188] A. G. Howard. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. Retrieved from <https://arxiv.org/abs/1704.04861>
- [189] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, Vol. 33.
- [190] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Predicting predictions from training data. In *ICML*.
- [191] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020).
- [192] Mahmut Kaya and Hasan Şakir Bilge. 2019. Deep metric learning: A survey. *Symmetry* 11, 9 (2019).
- [193] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*. PMLR, 5583–5594.
- [194] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. ViLT: Vision-and-language transformer without convolution or region supervision. In *ICML*.
- [195] Thomas N. Kipf and Max Welling. 2022. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [196] Sylwester Kłoczek, Łukasz Maziarka, Maciej Wołczyk, Jacek Tabor, Jakub Nowak, and Marek Śmieja. 2019. Hyper-network functional image representation. In *Artificial Neural Networks and Machine Learning—ICANN'19: Workshop and Special Sessions*.

- [197] Alex Krizhevsky, Geoffrey Hinton, and others. 2009. Learning multiple layers of features from tiny images. (2009).
- [198] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *NeurIPS*.
- [199] Brian Kulis et al. 2013. Metric learning: A survey. *Found. Trends Mach. Learn.* 5, 4 (2013).
- [200] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2019. Improved precision and recall metric for assessing generative models. In *NeurIPS*.
- [201] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*.
- [202] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A ConvNet for the 2020s. In *CVPR*.
- [203] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *CVPR*.
- [204] Muhammad Arslan Manzoor, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang. 2023. Multimodality representation learning: A survey on evolution, pretraining and its applications. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 3 (2023), 1–34.
- [205] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The penn treebank. *Comput. Linguist.* 19, 2 (6 1993), 313–330.
- [206] Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *ACL*. 3428–3448. <https://doi.org/10.18653/v1/P19-1334>
- [207] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. 2021. Modulated periodic activations for generalizable local functional representations. In *ICCV*.
- [208] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *ICLR*.
- [209] Francesco Mezzadri. 2007. How to generate random matrices from the classical compact groups. *Not. Am. Math. Soc.* 54, 5 (2007).
- [210] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, Vol. 2011. Granada, 4.
- [211] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
- [212] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. 2020. Designing network design spaces. In *CVPR*.
- [213] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP’19*. 3982–3992. <https://doi.org/10.18653/v1/D19-1410>
- [214] Jeanne Salle, Louis Jalouzet, Nur Lan, Emmanuel Chemla, and Yair Lakretz. 2024. What makes two models think alike? arXiv:2406.12620. Retrieved from <https://arxiv.org/abs/2406.12620>
- [215] Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. 2023. ModelDiff: A framework for comparing learning algorithms. In *ICML*.
- [216] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*. <https://doi.org/10.18653/v1/P18-1238>
- [217] K. Simonyan and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.
- [218] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. In *NeurIPS*.
- [219] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP’13*, David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (Eds.). Association for Computational Linguistics, 1631–1642.
- [220] Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, and Tom Goldstein. 2022. Can neural nets learn the same model twice? Investigating reproducibility and double descent from the decision boundary perspective. In *CVPR*.
- [221] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. 2015. Striving for simplicity: The all convolutional net. In *ICLR (Workshop Track)*. Retrieved from <http://lmbweb.informatik.uni-freiburg.de/Publications/2015/DB15a>
- [222] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*.
- [223] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. 2019. MnasNet: Platform-aware neural architecture search for mobile. In *CVPR*.
- [224] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*.

- [225] Tobias Uelwer, Jan Robine, Stefan Sylvius Wagner, Marc Höftmann, Eric Upschulte, Sebastian Konietzny, Maike Behrendt, and Stefan Harmeling. 2023. A survey on self-supervised representation learning. arXiv:2308.11455. Retrieved from <https://arxiv.org/abs/2308.11455>
- [226] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, Vol. 30.
- [227] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- [228] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 353–355. <https://doi.org/10.18653/v1/W18-5446>
- [229] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. 2023. All in one: Exploring unified video-language pre-training. In *CVPR*.
- [230] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Trans. Pattern Anal. Mach. Intell.* 46, 8 (2024), 5362–5383. <https://doi.org/10.1109/TPAMI.2024.3367329>
- [231] Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT'18, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. <https://doi.org/10.18653/v1/N18-1101>
- [232] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. 2016. Revisiting semi-supervised learning with graph embeddings. In *ICML*.
- [233] Jiaxuan You, Rex Ying, and Jure Leskovec. 2019. Position-aware graph neural networks. In *ICML*.
- [234] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2020. GraphSAINT: Graph sampling based inductive learning method. In *ICLR*.
- [235] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *CVPR*.

Appendices

A Overview of Notations and Basic Definitions

A.1 Notations

Within the notations in this survey, we use a few conventions. Sets are usually denoted with uppercase calligraphic letters, such as $\mathcal{N}, \mathcal{O}, \mathcal{P}$. Matrices $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, $n_1, n_2 \in \mathbb{N}$ are always denoted with bold uppercase letters, whereas vectors $\mathbf{v} \in \mathbb{R}^n$, $n \in \mathbb{N}$ are denoted with bold lowercase letters. General scalar variables $a \in \mathbb{R}$ are usually denoted with regular lowercase letters, whereas specific constants, such as the number of classes C in a classification task, or the dimension of representations D , are denoted with uppercase letters. Specific lowercase variables are reserved, such as m for model similarity measures or f for layer functions of neural networks. All of these fixed variables are given in Table 4, all other variables are excluded there.

A.2 Norms and Inner Products for Matrices

Here, we briefly describe the Frobenius and nuclear norm that are used in some representational similarity measures.

Frobenius Norm. On the vector space of all matrices in $\mathbb{R}^{n \times d}$, $n, d \in \mathbb{N}$, the *Frobenius inner product* is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_{i=1}^n \sum_{j=1}^d \mathbf{A}_{i,j} \mathbf{B}_{i,j} = \text{tr}(\mathbf{A}^T \mathbf{B}). \quad (\text{A.1})$$

This inner product induces the *Frobenius norm*, which for $\mathbf{A} \in \mathbb{R}^{n \times d}$ is defined as

$$\|\mathbf{A}\|_F = \sqrt{\langle \mathbf{A}, \mathbf{A} \rangle_F} = \sqrt{\sum_{i=1}^n \sum_{j=1}^d |\mathbf{A}_{i,j}|^2} = \sqrt{\text{tr}(\mathbf{A}^T \mathbf{A})} = \sqrt{\sum_{i=1}^{\min(n,d)} \sigma_i^2}, \quad (\text{A.2})$$

with σ_i denoting the i th singular value of \mathbf{A} . The Frobenius norm is invariant to orthogonal transformations.

Table 4. Overview of Notations

| | |
|----------------------------|---|
| f, f' | Neural networks |
| $f^{(l)}, f'^{(l')}$ | Layer l/l' of neural networks f, f' |
| L, L' | Total number of layers in f, f' |
| D, D' | Number of neurons of a layer |
| N | Number of inputs |
| C | Number of classes in a classification task |
| m | Similarity measure |
| q | Quality function |
| \mathbf{y} | Vector of ground-truth labels |
| \mathbf{X} | $N \times p$ matrix of N inputs |
| \mathbf{R}, \mathbf{R}' | $N \times D/N \times D'$ representation matrices |
| \mathbf{O}, \mathbf{O}' | $N \times C$ output matrices |
| \mathbf{S}, \mathbf{S}' | Representational Similarity Matrices |
| \mathcal{R}, \mathcal{O} | Sets of representation/output matrices |
| \mathcal{T} | Group of linear transformations |
| $\sim_{\mathcal{T}}$ | Equivalence up to transformations from \mathcal{T} |
| $O(D)$ | Group of orthogonal transformations |
| $GL(D, \mathbb{R})$ | Group of invertible matrices in $\mathbb{R}^{D \times D}$ |
| $\mathcal{N}_R^k(i)$ | Set of k nearest neighbors of i in R |
| \top | Transpose of a matrix/vector |
| $\mathbf{1}_n$ | Vector of n ones |
| \mathbf{I}_n | Identity matrix of size $n \times n$ |
| \mathbf{H}_n | Centering matrix of size $n \times n$ |
| $\mathbb{1}$ | Indicator function |

Nuclear Norm. Similarly to the Frobenius norm, one can define the nuclear norm in terms of the singular values of a matrix

$$\|\mathbf{A}\|_* = \sum_{i=1}^{\min(n,d)} \sigma_i. \quad (\text{A.3})$$

A.3 Similarity Functions for RSMs

When analyzing representational similarity, instancewise similarity functions $s : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, $n \in \mathbb{N}$ are often needed, in particular for RSM-based measures (here $n = D$). As noted in Section 3.3, they further strongly impact which groups of transformations these measures are invariant to. In the following, we provide a brief overview of common similarity functions, where we always assume two input vectors $\mathbf{v}, \mathbf{v}' \in \mathbb{R}^n$ to be given. We also provide an overview of the invariances that they induce on RSM-based measures in Table 5.

- *Euclidean Distance.* This well-known distance function is defined as

$$\|\mathbf{v} - \mathbf{v}'\|_2 = \sqrt{\sum_{i=1}^n (v_i - v'_i)^2}. \quad (\text{A.4})$$

This function satisfies the properties of a distance metric.

- *Cosine Similarity.* The cosine similarity between two vectors is defined as

$$\text{cos-sim}(\mathbf{v}, \mathbf{v}') = \frac{\mathbf{v}^\top \mathbf{v}'}{\|\mathbf{v}\|_2 \|\mathbf{v}'\|_2}. \quad (\text{A.5})$$

It is bounded in the interval $[-1, 1]$, with $\text{cos-sim}(\mathbf{v}, \mathbf{v}') = 1$ indicating that both vectors point in the exact same direction, and $\text{cos-sim}(\mathbf{v}, \mathbf{v}') = 0$ indicating orthogonality.

Table 5. Overview of Instancewise Similarity Functions

| Function | Induced Invariances | | | | | | Metric |
|---------------------|---------------------|----|----|-----|----|----|--------|
| | PT | OT | IS | ILT | TR | AT | |
| Euclidean distance | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |
| Cosine similarity | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |
| Linear kernel | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| RBF kernel | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Pearson correlation | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ |

- *Linear Kernel*. This kernel is defined as

$$K(\mathbf{v}, \mathbf{v}') = \mathbf{v}^\top \mathbf{v}'. \quad (\text{A.6})$$

When \mathbf{v}, \mathbf{v}' have unit norm, this measure is equivalent to cosine similarity. $K(\mathbf{v}, \mathbf{v}') = 0$ indicates that vectors are orthogonal to each other. The linear kernel is not bounded.

- *Radial Basis Function Kernel*. The RBF kernel is defined as

$$K_\sigma(\mathbf{v}, \mathbf{v}') = \exp\left(-\frac{\|\mathbf{v}-\mathbf{v}'\|_2^2}{2\sigma^2}\right), \quad (\text{A.7})$$

where $\sigma \in \mathbb{R}$ is a free parameter. With a range of $[0, 1]$, $K_\sigma(\mathbf{v}, \mathbf{v}') = 1$ indicates maximum similarity, a value of zero indicates minimal similarity.

- *Pearson Correlation*. Letting $\bar{\mathbf{v}} = \frac{1}{n} \sum_{i=1}^n \mathbf{v}_i$ denote the average value of the vector \mathbf{v} , the Pearson correlation coefficient is defined as

$$r(\mathbf{v}, \mathbf{v}') = \frac{\sum_{i=1}^n (\mathbf{v}_i - \bar{\mathbf{v}}_i)(\mathbf{v}'_i - \bar{\mathbf{v}}'_i)}{\sqrt{\sum_{i=1}^n (\mathbf{v}_i - \bar{\mathbf{v}}_i)^2} \sqrt{\sum_{i=1}^n (\mathbf{v}'_i - \bar{\mathbf{v}}'_i)^2}}. \quad (\text{A.8})$$

This function is bounded in the interval $[-1, 1]$, with $r(\mathbf{v}, \mathbf{v}') = 1$ indicating perfect correlation, and $r(\mathbf{v}, \mathbf{v}') = 0$ no correlation at all. When \mathbf{v}, \mathbf{v}' are mean-centered, i.e., $\bar{\mathbf{v}} = \bar{\mathbf{v}}' = 0$, this function is equivalent to cosine similarity.

A.4 Intertwiner Groups

Godfrey et al. [47] introduced the concept of *intertwiner groups*, which they applied to analyze symmetries in neural network models. Formally, given an invertible activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, its corresponding intertwiner group is defined as

$$G_\sigma := G_{\sigma, D} = \{\mathbf{A} \in \text{GL}(D, \mathbb{R}) : \exists \mathbf{B} \in \text{GL}(D, \mathbb{R}) \text{ s.t. } \sigma \circ \mathbf{A} = \mathbf{B} \circ \sigma\}, \quad (\text{A.9})$$

where $\text{GL}(D, \mathbb{R})$ denotes the general linear group of invertible matrices in $\mathbb{R}^{D \times D}$. The corresponding group of transformations of neural representations is then defined as

$$\mathcal{T}_\sigma = \{\mathbf{R} \mapsto \mathbf{R}\mathbf{M} : \mathbf{M} \in G_\sigma\}. \quad (\text{A.10})$$

We highlight the case where $\sigma = \text{ReLU}$, which yields the group $\mathcal{T}_{\text{ReLU}}$, because Godfrey et al. [47] detail similarity measures that are invariant to transformations from $\mathcal{T}_{\text{ReLU}}$. G_{ReLU} consists of matrices of the form $\mathbf{P}\mathbf{D}$, where $\mathbf{P} \in \mathcal{P}$ is a permutation matrix and \mathbf{D} is a diagonal matrix with positive elements. Thus, invariance to $\mathcal{T}_{\text{ReLU}}$ implies invariance to permutations, and when assuming representations with normalized columns, one can, for instance, constrain the orthogonal Procrustes measure to be invariant to this group [47].

B Preprocessing of Representations

Next, we discuss techniques for normalization, adjusting dimensionality, and flattening of representations.

Normalization. Some similarity measures assume that the representations are normalized. For instance, it is commonly assumed that representations are mean-centered in the columns [73, 101, 155]. Mean-centering effectively constitutes a translation of the representations, which imposes the assumption that representations are equivalent under translations. In consequence, the corresponding measures are invariant toward translations. For such reasons, normalization methods should be used with caution, as they require the compared representations to be compatible with such assumptions.

In the following, we briefly discuss some commonly used normalization methods for representations or RSMs. To keep the broader scope, we consider normalization of matrices $\mathbf{M} \in \mathbb{R}^{n \times d}$, $n, d \in \mathbb{N}$. Then, a normalization can be considered as a mapping $\psi : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$. To simplify notation, in this context we apply the *centering matrix* \mathbf{H}_n , $n \in \mathbb{N}$, which is defined as $\mathbf{H}_n = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$,

- *Rescaling of Instances to Unit Norm.* Letting $\mathbf{D} = \text{diag}(\|\mathbf{M}_1\|_2, \dots, \|\mathbf{M}_n\|_2)$ denote the diagonal matrix of row lengths, this rescaling can be written as a transformation,

$$\mathbf{M} \mapsto \mathbf{D}^{-1} \mathbf{M}. \quad (\text{B.11})$$

This transformation preserves angles but alters Euclidean distances between vectors.

- *Rescaling of Columns to Unit Norm.* Letting $\mathbf{D} = \text{diag}(\|\mathbf{M}_{-,1}\|_2, \dots, \|\mathbf{M}_{-,d}\|_2)$ denote the diagonal matrix of column lengths, this rescaling can be written as a transformation

$$\mathbf{M} \mapsto \mathbf{M} \mathbf{D}^{-1}. \quad (\text{B.12})$$

This transformation preserves neither angles nor distances.

- *Rescaling of Matrix to Unit Norm.* This preprocessing rescales the whole matrix to unit norm:

$$\mathbf{M} \mapsto \frac{\mathbf{M}}{\|\mathbf{M}\|_F}. \quad (\text{B.13})$$

Like the previous rescaling, angles are preserved, but Euclidean distances are not.

- *Mean-Centering of Columns.* This normalization sets the column means to zero, while preserving their variance. It can be written as a transformation

$$\mathbf{M} \mapsto \mathbf{H}_n \mathbf{M}, \quad (\text{B.14})$$

which effectively constitutes a translation of the representations. Thus, it alters angles but preserves Euclidean distance between representations.

- *Double Mean-Centering.* This approach translates both rows and columns such that both row and column means equal zero. For any matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, $n, d \in \mathbb{N}$, double mean-centering in rows and columns can be defined as a transformation

$$\mathbf{M} \mapsto \mathbf{H}_n \mathbf{M} \mathbf{H}_d. \quad (\text{B.15})$$

This normalization is typically not applied directly to representations, as it would translate individual rows differently, and alter both Euclidean distance and angles between the row vectors.

Adjusting Dimensionality. Many of the representational similarity measures presented in Section 3 implicitly assume that the representations \mathbf{R}, \mathbf{R}' have the same dimensionality, i.e., $D = D'$. Thus, if $D < D'$, then some preprocessing technique must be applied to match the dimensionality. Two techniques have been recommended for preprocessing: zero-padding and dimensionality reduction, such as PCA [155, 183]. When zero-padding, the dimension D of representation \mathbf{R} is inflated by appending $D' - D$ columns of zeros to \mathbf{R} . PCA conversely reduces the

dimension of the representation R' by removing the $D' - D$ lowest-information components from the representation.

Flattening. Representational similarity measures assume matrices $R \in \mathbb{R}^{N \times D}$ as input. However, some models such as CNNs produce representations of more than two dimensions, making them incompatible with these measures. In such a case, representations have to be flattened, taking into account model-specific properties of representations. For example, representations from CNNs usually have the form $R \in \mathbb{R}^{N \times h \times w \times c}$, where h and w denote height and width of the feature maps and c the number of channels. Directly flattening these representations into matrices $R \in \mathbb{R}^{N \times hwc}$ would yield a format in which permuting the features would disregard the spatial information in the original feature map, which may be undesirable. To avoid this issue, flattening CNN representations into matrices $R \in \mathbb{R}^{Nhw \times c}$ yields representations where permutations only affect the channels [155]. However, when comparing two models f, f' , their flattened representations are only compatible if the height and width of both models match or a feature map is upsampled, as the number of rows in the resulting matrices must match. Further, computational cost of a similarity measure may be affected by the new effective numbers of features and inputs in the flattened representation.

C Analyses of Similarity Measures

This section gives an overview of analyses of similarity measures that study the relation between representational and functional similarity, what kind of representations similarity measures can distinguish, and how the scores are influenced by the given inputs. A summary of the comparative evaluations is shown in Table 3.

C.1 Correlation between Functional and Representational Measures

There has only been little work that investigates the relationship between representational and functional similarity. Most prominently, Ding et al. [34] studied on BERT [33] and ResNet [187] models whether diverging functional behavior correlates with diverging representational similarity. To that end, they induced functional changes on the given models, such as varying training seeds, removing principal components of representations at certain layers, or applying out-of-distribution inputs, and investigated whether observed changes in accuracy on classification tasks correlate with changes in representational similarity as measured by CKA (Equation (28)), PWCCA (Equation (14)), and orthogonal Procrustes (Equation (15)). They observed that orthogonal Procrustes generally correlates with changes in functional behavior to a higher degree than CKA and PWCCA. Further, CKA appeared much less sensitive to removal of principal components of representations than orthogonal Procrustes and PWCCA—it still indicated high similarity between the original and the modified representation when the accuracy of the model has already dropped by over 15%. GULP (Equation (32)) was later benchmarked using the same protocol and found to perform similarly to CKA and Procrustes, although it relied on good selection of regularization strength [17].

A similar analysis was conducted by Hayne et al. [57], who induced functional changes by deleting neurons in the linear layers of CNNs that were trained on ImageNet [179]. They reported that orthogonal Procrustes and CKA correlate more with functional similarity than CCA measures. Barannikov et al. [8] further compared disagreement (Equation (51)) of models with CKA and RTD (Equation (41)) scores. CKA correlated to a lower degree than RTD. Boix-Adsera et al. [17] correlated representational similarity with mean squared difference between outputs of regression models that were trained on the representations with random labels. They found that GULP correlated better than CCA-based measures and CKA. The recent ReSi benchmark [70] correlated over

20 representational similarity measures with accuracy, disagreement, and JSD. They used representations from vision, text, and graph models across multiple datasets. No measure consistently outperformed others across these tests.

Davari et al. [178] pointed out how CKA is sensitive to manipulations of representations that would not affect the functional similarity of the underlying models. For instance, they showed that one can alter the CKA of two identical representations to almost zero by translating the representation of a single instance in one of the copies, without affecting the separability of the representations with respect to their class. Further, they could modify representations of multiple layers to obtain prespecified CKA scores between them, while leaving functional similarity almost unaffected. Similar results were also reported by Csizsárik et al. [30].

C.2 Discriminative Abilities of Representational Similarity Measures

Invariances of representational similarity measures indicate which representations are considered equivalent. However, measures have practical differences in distinguishing representations, which have been assessed in numerous works.

Morcos et al. [101] tested the robustness of CCA-based measures (see Section 3.1) to noise in representations. They argued that measures should identify two representations as similar if they share an identical subset of columns, next to a number of random noise dimensions. In their experiments, they found that PWCCA is most robust in indicating high similarity, even if half of the dimensions are noise. By comparison, mean CCA was the least robust.

A number of works [24, 73, 116, 125] have explored the ability of representational similarity measures to match corresponding layers in pairs of models that only differ in their training seed. For instance, given two model instantiations and comparing layer five in one instantiation with all layers from the other model, the similarity with layer five from the other instantiation should be the highest. No measure clearly outperformed other measures consistently.

Shahbazi et al. [125] tested whether representations obtained by sampling a low number of dimensions from a baseline representation yield high similarity with the baseline or other low-dimensional samples. They compared CKA (Equation (28)), Riemannian distance (Equation (33)), RSA (Equation (27)), and RSM norm difference (Equation (26)) on a neuroscience dataset, with sampled dimensions varying between 10 and 50. For higher dimensions, all measures assigned high similarity between the samples and the baseline. For low dimensions, only Riemannian distance consistently assigned high similarity between the sample and its original representation. Other measures yielded lower similarities, yet CKA gave better results than RSA and the norm-based measure.

Barannikov et al. [8] used synthetic data patterns to test the ability of RTD (Equation (41)) to discriminate between topologically different data. They generated data consisting of increasing amounts of clusters, which were arranged circularly in two-dimensional space, and argued that the similarity between the dataset of one cluster and datasets with more clusters should decrease with increasing number of clusters. The rank correlation between similarity score of a measure and number of clusters in the data was perfect for RTD, whereas CKA, SVCCA, and IMD (Equation (40)) had relatively low correlations.

Boix-Adsera et al. [17] assumed that models of similar architecture have similar representations. Hence, they clustered ImageNet-trained models based on pairwise representational similarity and measured the quality of the resulting clusters. CKA, orthogonal Procrustes (Equation (15)), and GULP (Equation (32)) all allowed for good clustering in general; CCA-based measures tended to perform worse in comparison. With optimized regularization strength λ , GULP overall yielded the best clustering among these measures. Further, GULP clustered well even for inputs from other datasets.

Tang et al. [141] argued that models trained from two similar datasets, such as CIFAR-10 and CIFAR-100 [197], should be more similar compared to models trained on dissimilar datasets, that for instance do not contain natural images. In their experiments, they compared CKA and NBS with respect to this desideratum, but results were inconclusive.

Rahamim and Belinkov [116] tested whether representations of text in different languages for a fixed model are more similar than representations of two random texts. Similarly, they evaluated whether the representation of an image is most similar to the representation of the true caption compared to captions of other images. In both cases, ContraSim (Equation (24)), which was specifically trained for the respective task, outperformed CKA.

Finally, the ReSi benchmark [70] proposed four tests that evaluate the discriminative abilities of measures. In three of these tests final-layer representations of models with different behavior need to be distinguished. These behavior differences stem from training with varying amount of random labels, shortcut features, or augmentation. The fourth test correlated similarity in layer depth with representational similarity. All tests are implemented over models from the vision, language, and graph domains across multiple datasets. Initial results from the benchmark indicate that there is no similarity measure that performs well over all tests and domains.

C.3 Influence of Inputs

Another issue studied in literature is the impact of the inputs X on similarity scores. For popular functional similarity measures, it is well known that similarity of outputs is confounded by the accuracy of the models, the number of classes and the class distribution [14, 43, 69, 167, 173]. Similar confounding effects also exist with respect to representational similarity measures. In the following, we discuss corresponding results.

First, Cui et al. [177] argued that similarity between input instances leads to similarity of their representations in early layers, as the extracted low-level features—even if they are different overall—cannot clearly distinguish between instances. Thus, RSMs mirror the pairwise similarities of the inputs, which leads to high similarity estimates between models that may actually be dissimilar. This was demonstrated by showing that two random neural networks can obtain higher RSA (Equation (27)) and CKA (Equation (28)) scores than a pair of networks trained for the same task. To alleviate this problem, they proposed a regression-based approach to de-confound the RSMs.

Second, it was shown that representational similarity measures can be confounded by specific input features. Dujmović et al. [181] compared a model trained on standard image data to models trained on modified images. The modified images contained a class-leaking pixel to allow models to learn a shortcut for classification. The locations of the leaking pixels affected representational similarity between the models, measured by RSA. Similarly, Jones et al. [64] found that feature co-occurrence in inputs may lead to overestimation of model similarity by CKA. Different input features may co-occur in the data used to compute representations, but models may use these features to different extents. For example, on a high level, the features “hair” and “eyes” co-occur in images of human faces, but one model may only use the hair to compute its representations, whereas the other model may only use the eyes feature. They showed that CKA scores ignore the difference in feature use with an image inversion approach: Using data synthetically generated to produce the same representations in one model, similarity to the other model dropped drastically as feature co-occurrences were eliminated.

Third, the number of input instances N may influence similarity scores. Williams et al. [155] compared two CNNs trained on CIFAR-10. The representations were computed from the test data with varying sample size N . The similarity between the two CNNs in terms of the Angular Shape Metric (Equation (16)) generally decreased with increased ratio N/D , before a stable score was

reached that did not change with more inputs. When constraining the measure to permutation invariance, a lower ratio N/D was sufficient to achieve a stable similarity score.

Finally, the effect of the choice of inputs was studied by Brown et al. [172]. Using inputs from different data distributions significantly affected similarity scores of CKA and orthogonal Procrustes. However, similarity between models when giving in-distribution data was significantly correlated with similarity when given out-of-distribution data. The extent of correlation heavily depends on the specific dataset.

D Details on Evaluation of Representational Similarity Measures

In Table 3, we show the rankings of representational similarity measures from different tests in literature, which we describe in Appendix C. Most of the tests, however, considered multiple variants where, for instance, datasets and models have been varied. To obtain the single rank that is presented in Table 3, we first created rankings for each test variant. We then averaged these ranks for each measure and finally assigned the ranks as depicted in Table 3 based on these averages. We note that these aggregated results do not highlight the considerable variance in performance that often occurred across test variants. For example, k -NN Jaccard similarity is the best measure on average for the JSD correlation test in the ReSi benchmark, but across all model and data variants its rank varies between 1 and 18. Further, these ranks do not indicate statistically significant differences. The ranks should only be interpreted as a general direction of performance. Thus, we generally recommend looking into the study that a test originated from to obtain more nuanced insights regarding the applicability of a measure for specific application scenarios.

In the following, we provide more detailed descriptions regarding how we determined and (if necessary) aggregated ranks for each of the listed tests. To provide some further orientation, we give an overview of the models and datasets that were considered in each test in Table 6.

Accuracy Correlation. From the experiments conducted by Boix-Adsera et al. [17], we aggregated the results depicted in Figures 22–24 in their appendix, where we computed individual rankings based on the correlation measured by Spearman’s rho. As for the rank of GULP with optimized λ , we always chose the best result across all values of λ in each individual test. For the analysis by Ding et al. [34], we aggregated ranks over all tests with respect to Spearman correlation, as they depicted in Table 1. Since PWCCA was not applicable in the vision tests, we ranked it last in the tests from these domains. From the experiments by Hayne et al. [57], we used the data published in their code repository. We first averaged the correlation values over all layers and then created separate rankings per model. Regarding the ReSi benchmark [70], we considered and aggregated all results for test 1 (correlation to accuracy difference) as presented in Appendix B, where we ranked based on the reported Spearman correlation.

Disagreement Correlation. From the experiments by Barannikov et al. [8], we considered the results reported in Tables 1, 2 and 4, where RTD always outperformed CKA. Regarding ReSi [70], we considered and aggregated all results for test 2 (correlation to output difference) as presented in Appendix B, where we ranked based on the reported Spearman correlation of representational similarity measures with disagreement.

JSD Correlation. Again, we considered and aggregated all results for test 2 (correlation to output difference) of ReSi [70] as presented in Appendix B, where we ranked based on the reported Spearman correlation of representational similarity measures with JSD.

Squared Error Correlation. We allocated ranks based on the Spearman correlations reported in Reference [17, Figure 4] that was averaged over the two given regularization strengths of the given linear predictors. Regarding the rank of GULP with optimized λ , we chose the best result across all values of λ at each regularization strength.

Noise Addition. To construct the ranks for the experiments by Morcos et al. [101], we considered the areas under the curves as presented in Figure 2. They only considered one model and dataset, so we did not aggregate ranks.

Layer Matching. Chen et al. [24] present the effect of hyperparameters on matching accuracy in Figure 3. We rank the measures based on the accuracy with respect to the optimal hyperparameters mentioned in the text (degree 5 and graph size 50) for the three used architectures. For Kornblith et al. [73], we rank measures based on the matching accuracies with respect to both CNNs (Table 2) and Transformers (Table F.1). From the experiments by Rahamim and Belinkov [116], we consider the results depicted in Table 1, where we rank each combination of encoder training set with representation dataset as a variant (separately for each domain). For Shahbazi et al. [125], we rank measures based on the mean matching accuracy as reported in their Figure 13.

Dimension Subsampling. We considered and aggregated the results from Shahbazi et al. [125] as depicted in Figures 5 and 6. For these individual experiments, ranks were, again, determined by the depicted areas under the curves. For RSA, we used the Spearman curve, and RSM Norm Difference corresponds to the Euclidean curve.

Cluster Count. We considered and aggregated results from both experiments with synthetic clusters and rings, where we ranked measures according to the Kendall's τ rank correlation with the number of clusters and rings, respectively, as reported by Barannikov et al. [8, Section 3.1].

Architecture Clustering. From the experiments by Boix-Adsera et al. [17], we consider the results depicted in Figure 16, where we consider the pretrained and untrained models as different variants and rank measures by their average standard deviation ratio.

Multilingual. From the multilingual benchmark by Rahamim and Belinkov [116], we considered the results depicted in Tables 2, 4, and 5, where, in every test variant and for each probing layer, ContraSim had higher accuracy than CKA.

Image Caption. From the image caption benchmark by Rahamim and Belinkov [116], we considered the results depicted in Figure 5 and Table 3 and 6, where, again, ContraSim had higher accuracy than CKA in all test variants.

Shortcut Affinity. We considered and aggregated all results for test 4 (shortcut affinity) of ReSi [70] as presented in Appendix B, where we ranked all measures based on the reported AUPRC scores.

Augmentation. We considered and aggregated all results for test 5 (augmentation) of the ReSi benchmark [70] as presented in Appendix B, where we ranked all measures based on the reported AUPRC scores.

Label Randomization. We considered and aggregated all results for test 3 (label randomization) of the ReSi benchmark [70] as presented in Appendix B, where we ranked all measures based on the reported AUPRC scores.

Layer Monotonicity. We considered and aggregated all results for test 6 (layer monotonicity) of the ReSi benchmark [70] as presented in Appendix B, where we ranked all measures based on the reported Spearman correlation.

E Neural Network Similarity beyond This Survey

In this survey, we reviewed representational similarity measures that can compare representations from two different models that use the same inputs, and functional similarity measures that compare models in (multiclass) classification contexts. Beyond the scope of this survey, there are other views on neural network similarity and application contexts, which we briefly discuss here.

Functional Similarity for Non-Classification Tasks. Although we focus on functional similarity with respect to classification, many of the functional similarity measures can be used for or directly transferred to other downstream tasks. In particular, if a suitable performance measure is given, performance-based measures can be used in any other context. This is also the case for gradient-based and stitching measures if white-box access to the models is given, and, in case of gradient-based measures, adversarial examples can be constructed for the given context. Soft and hard prediction-based measures, conversely, are limited to tasks where outputs are assigned discrete labels. For regression, one could consider binning outputs to obtain discrete labels. Further, there are specialized measures of agreement of continuous outputs [46, 143]. Finally, if output is structured, e.g., text or image generation, functional similarity becomes more difficult as outputs do not share universally identical semantics as in classification tasks. For example, generated images may have differences that are not perceivable to the human eye and thus could be considered equivalent. This equivalence could lead to considering the invariances of functional similarity measures. The evaluation of these kinds of models, including comparison of outputs to a human reference, was studied in prior surveys [18, 22].

Alternative Notions of Neural Network Similarity. Aside from representational and functional similarity measures, there are several other notions of similarity that have been used to compare neural networks. Some of these approaches are applicable for specific types of neural networks. For instance, *visualizations* have emerged as a popular tool to analyze CNN similarity, although not limited to them. Approaches include the visualization of decision regions [220], neuron activations [169], or reconstructed images [203]. Chen et al. [174] proposed a method to compare *weights* of convolutional layers. For language models, *probing* [170] has become a popular approach. The idea behind probing is to compare the extent to which representations of models trained for a specific task such as sentiment analysis can also be used to predict related concepts such as part of speech.

There are also more universal approaches. For instance, Wang et al. [151] and Guth and M  nard [184] proposed methods to compare the *weight matrices* of neural networks. Further, one could also consider the impact of inputs, as done by Shah et al. [215]. They utilize the concept of *datamodels* [190], which aim to explain predictions in terms of which data samples were used in training. Using that approach, they measure similarity in neural networks by comparing the influence that data points have on individual predictions. Salle et al. [214] considered the extent to which differences in meta-features, such as part of speech or tense for text models, predict differences in instance representations, and compared different models based on the importance of such features for the prediction. Finally, measures that compare representations that are derived from different sets of inputs but mapped into the same vector space, e.g., by coming from the same model, were proposed in the context of evaluating generative adversarial networks [168, 200] and metric learning [192, 199].

Representational Similarity for Training Neural Networks. Optimizing representations for high or low similarity during model training is a reoccurring theme across deep learning, e.g., in knowledge distillation [49] or fields that use contrastive representation learning [182, 191, 204, 225, 230].

The approaches to assessing similarity in this context are different from the similarity measures in this survey. First, differentiability and computational efficiency become important properties to enable gradient descent-based optimization. Second, and more importantly, in these processes it is often assumed that representations lie in the same representation space [192, 193, 199], and similarity is often evaluated on the instance level rather than on full representation matrices [194, 213, 229]. Alternatively, if the representations come from different models such as in multimodal representation learning, their mapping into the joint space can be trained together with the rest

of the system [229]. Hence, unless invariances of the similarity measure should be used to make optimization more flexible, invariances are not important. While these approaches are useful for training and aggregating representations at fixed layers of models, they do not generalize to post-training analysis of neural networks.

Distinction to Functional Representations. This survey covers representational and functional similarity measures. The terms *representational* and *functional* should not be confused with *functional representations*. In contrast to our work, which is about comparing neural networks, work on functional representations is about training neural networks to represent continuous functions that are only known via samples at discrete points [196, 207, 218].

F Orthogonal Procrustes and Dimensionality

To demonstrate how the similarity scores of a measure may be influenced by external factors such as dimensionality, we plot values of the orthogonal Procrustes measure over varying dimension in Figure 5.

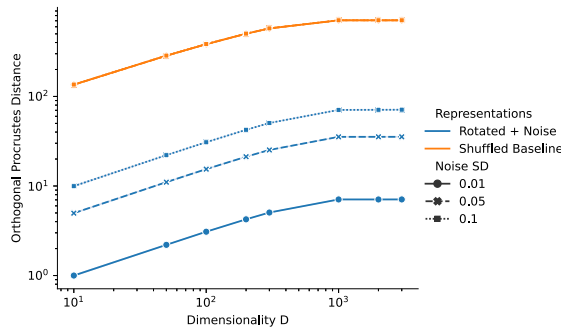


Fig. 5. Mean orthogonal Procrustes scores between two matrices over increasing dimensionality with varying noise level. The matrices have $N = 1,000$ rows. Shuffled Baseline refers to the score between two effectively unrelated matrices, a row-wise shuffled copy of the representation matrix and the original, similar to Kriegeskorte et al. [74]. The baseline is unrelated to the noise level. Scores increase until the number of dimensions matches the number of inputs ($N = D$) and then stays flat. While $N > D$, the relation between the similarity score and the dimensionality follows a power law, as shown by the linear relation in the log-log plot. The standard deviation is too small to be visible. The same trend can be observed with other N (not shown).

We compare two synthetic representation matrices: the first matrix is a random matrix with entries drawn from a standard normal distribution, and the second matrix is generated by multiplying the first matrix with an orthogonal matrix that was randomly drawn from the Haar distribution as implemented by `scipy`⁴ [209], with added noise, that is again drawn from a normal distribution. These matrices have $N = 1,000$ rows and varying dimension $D \in \{10, 50, 100, 200, 300, 1,000, 2,000, 3,000\}$. This matrix generation process is repeated 10 times for each value D , and we report the mean orthogonal Procrustes distance resulting from these matrix pairs. In addition, we create a baseline similarity score by permuting the rows of a copy of the original representation matrix and compare it to the original representation matrix, similarly to the technique proposed by Kriegeskorte et al. [74]. We compute the baseline scores by shuffling the rows 10 times for each representation pair, again reporting the mean.

The code to this experiment is available on GitHub.⁵

⁴https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ortho_group.html

⁵https://github.com/mklabunde/survey_measures

G Transformations for Figure 2

In Figure 2, the AT, ILT, and TR invariances use

$$\mathbf{A} = \begin{bmatrix} 0.68 & 0.05 \\ 0.22 & 0.18 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 1.2 & -1.6 \end{bmatrix}.$$

The illustrations of the OT, PT, and IS invariances use the following parameter values in their respective transformations:

$$\mathbf{Q} = \begin{bmatrix} -0.87 & 0.5 \\ 0.5 & 0.87 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad a = 2.$$

The transformation \mathbf{Q} corresponds to rotating the representation by 120° and reflecting across the 15° axis. The permutation \mathbf{P} effectively swaps the axes in the coordinate system.

Received 9 August 2023; revised 28 February 2025; accepted 1 April 2025