LEARNING TO COMPARE NEURAL REPRESENTATIONS FROM PARAMETERIZED COMPUTATION GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Comparing internal representations across trained models is important for identifying corresponding layers and selecting components for model reuse. Yet, direct comparisons at inference require evaluating candidate models on known reference inputs and collecting their internal activations. In this paper, we study whether internal representation similarity measured across independently trained models can instead be learned to predict in unseen models from their *parameterized computations*. We introduce a framework that applies a graph metanetwork for mapping each internal site to an embedding derived from its upstream computations while accounting for parameter symmetries. We analyze the structural properties of our estimators and derive conditional guarantees for layer retrieval and stitching candidate selection. Empirically, we find that the learned predictors support relation estimation and low-regret layer retrieval. They also select useful stitching connections while not being trained on stitching outcomes. These results show that representational relations learned across a collection of trained models can generalize to new computations, and that the predicted relations can guide candidate selection before direct measurement or adapter fitting for functional evaluations.

1 INTRODUCTION

Finding corresponding representations across trained neural networks (Bau et al., 2019; Huh et al., 2024; Gurnee et al., 2024) is an important scientific task for understanding shared internal structure across independently trained models (Li et al., 2015; Morcos et al., 2018; Raghu et al., 2017; Kornblith et al., 2019) and for identifying components that can be reused across networks (Yosinski et al., 2014; Bansal et al., 2021; Csiszárik et al., 2021; Moschella et al., 2023). Both goals motivate searching within and across a collection of trained models: given an internal representation in one model, which sites provide useful counterparts for analysis or potential connection points for model composition (Maiorca et al., 2023; Cannistraci et al., 2024; Pan et al., 2023; Yang et al., 2022; Hu et al., 2024; Wang et al., 2023)? These correspondences cannot be determined by layer indices or tensor shapes (Nguyen et al., 2021; Khosla & Williams, 2024; Singh & Jaggi, 2020), since networks may differ in depth and width, and each representation depends on the trained computation that produces it (Yurochkin et al., 2019; Tatro et al., 2020).

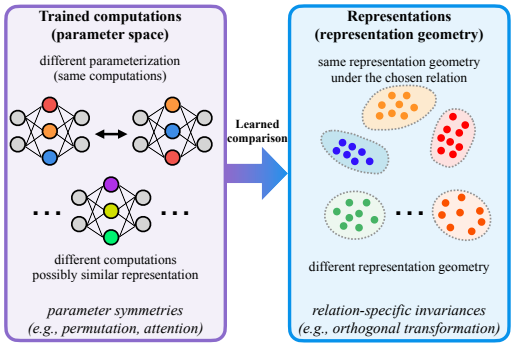

Figure 1: Learning a connection between trained computations and representation geometry. A representation site is determined by its upstream computation, but symmetry-related parameterizations can induce equivalent computations. Representational correspondence can also extend beyond these parameter symmetries: different computations may produce the same representation geometry under a chosen relation.

To compare internal representations at different sites across models, i.e., layers or other locations at which activations are measured, multiple criteria exist (Klabunde et al., 2025b; Kriegeskorte et al., 2008; Diedrichsen & Kriegeskorte, 2017; Ding et al., 2021). These methods emphasize different aspects of representational similarity, including response geometry, neighborhood structure, and pre-

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

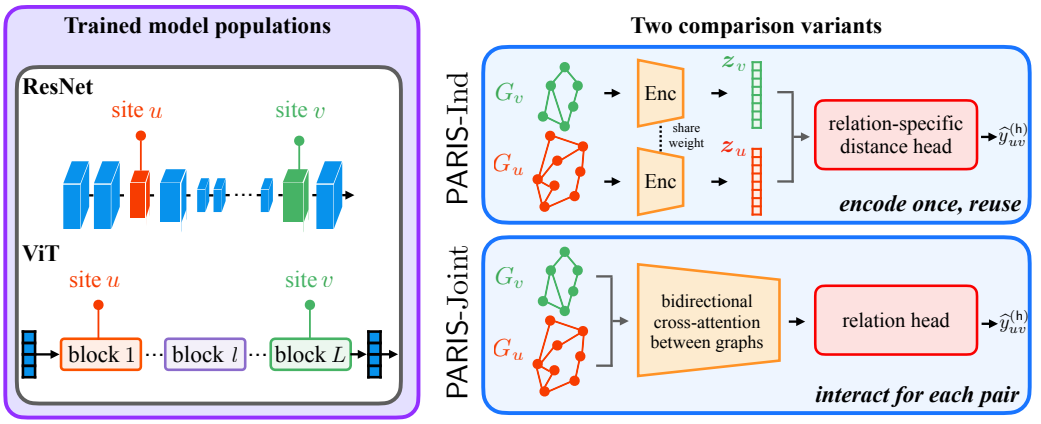

Figure 2: Overview of PARIS. We consider a population of trained models. Each site is described by a graph of its upstream computation. PARIS-Ind encodes each site independently and compares reusable embeddings using relation-specific distance heads. PARIS-Joint allows the two site graphs to interact before predicting their relation. Both variants are trained with internal representation similarity and rank candidate sites in unseen models without collecting new base-model activations. Our metanetworks can guide the selection of sites and be used for adapter fitting.

dictive behavior (Morcos et al., 2018; Williams et al., 2021; Davari et al., 2023; Barannikov et al., 2022). For instance, CKA (Kornblith et al., 2019) compares response geometry through the alignment of centered kernel matrices, CKNNA (Huh et al., 2024) focuses on local structure by restricting kernel alignment to shared nearest neighbors, and readout-based methods compare representations through the disagreement of fitted predictors under a specified prediction protocol (Alain & Bengio, 2016; Harvey et al., 2024). Direct evaluation of these activation-based relations relies on responses to shared reference inputs. Responses from multiple sites can be collected during the same model traversal, and the responses or suitable comparison descriptors can be cached across comparisons. For previously unmeasured models, however, the corresponding responses still need to be acquired. We investigate a different use of these measurements: as supervision for learning how parameterized computations relate to the representations they produce. This leads to our central question:

> *Can we learn to predict representation similarity scores in unseen models across different sites using only their architecture and trained weights?*

Parameter symmetries (Hecht-Nielsen, 1990; Godfrey et al., 2022; Zhao et al., 2026) pose one source of challenge for this learning problem (see Fig. 1). Even when tensor shapes match, relabelings of hidden units can yield different parameter configurations that produce the same network function (Sussmann, 1992; Chen et al., 1993; Ainsworth et al., 2023). Distances computed directly in parameter space can conflate differences in learned computation with differences in coordinate choice (Phuong & Lampert, 2020; Vlačić & Bölcskei, 2021; Stock & Gribonval, 2023; Bona-Pellissier et al., 2023; Simsek et al., 2021). Representational correspondence can also extend beyond these parameter symmetries. For example, orthogonally rotating features preserves their inner-product geometry while changing the individual feature values (Kornblith et al., 2019). Different upstream computations may produce similar representation geometry without being connected by these transformations. The learning problem is thus both to recognize equivalent parameterizations and to characterize which differences in trained computations matter for representational correspondence.

To this end, we introduce *Parameter-based Analysis for Relational Inference of Sites* (PARIS), a framework for learning internal representation relations from the computations that produce them (see Fig. 2). To accommodate variation in architecture, we represent each site's entire upstream computation as a graph encoding its architecture, trained parameters, and inference state. To account for permutation symmetries, the encoding maps relabelings of hidden units and channels to graph relabelings while preserving parameter correspondences across successive layers. For attention modules, we encode the composed query-key and value-output operators, which are invariant to the changes of basis within their internal spaces. To learn representational correspondences beyond those induced by parameter symmetries, we train the graph metanetwork using representation sim-

ilarity between sites across independently trained models. At search time, it predicts these relations for unseen models from their site graphs, without requiring new base-model forward passes.

We consider two variants that differ in when information from the two sites is combined. *(i)* PARIS-Ind maps each site graph to a fixed embedding and compares embeddings using relation-specific distance heads. These embeddings can be computed once and reused across comparisons. *(ii)* PARIS-Joint uses cross-attention between the two site graphs before predicting their relation, allowing intermediate features of each site to interact with the other. Comparing the two variants tests whether such interactions improve relation prediction beyond what can be achieved with site embeddings.

Our contributions are fourfold: *(i)* we introduce the problem of predicting representation similarity between internal representations from their parameterized upstream computations, *(ii)* we develop graph-based models that account for parameter symmetries and compare two approaches: encoding each site independently into an embedding and processing pairs of sites jointly, *(iii)* we analyze the structural properties of both estimators and derive conditional bounds connecting prediction error to retrieval regret and stitching-candidate selection, and *(iv)* we evaluate how well our models predict comparison scores, retrieve layers, and select stitching candidates across model populations of varying widths and depths, showing that training on the representation similarity on parameterized computations can support candidate selection without using stitching outcomes as supervision.

2 Related work

Our work connects the comparison of internal representations across models with learning from neural network artifacts. We discuss the two topics here; App. A discusses extended related work.

Comparing and matching neural representations. Comparisons across independently trained networks have been used to study recurring features and shared representational structure (Li et al., 2015; Gurnee et al., 2024; Huh et al., 2024). Activation-based approaches compare shared subspaces using SVCCA and PWCCA (Raghu et al., 2017; Morcos et al., 2018), response geometry using CKA (Kornblith et al., 2019), or regularized linear prediction using GULP (Boix-Adsera et al., 2022). Model stitching evaluates compatibility by connecting parts of trained networks (Bansal et al., 2021; Csiszárik et al., 2021), while component reassembly methods select fragments from model collections (Yang et al., 2022). Direct evaluation of activation-based relations depends on responses to reference inputs. In this paper, PARIS addresses the complementary problem of predicting these relations from the *parameterized computations* that produce the representations.

Learning from neural network parameters. Learning from trained weights has been used to predict network accuracy and other model characteristics (Unterthiner et al., 2020; Schürholt et al., 2021). DWSNets and neural functional networks account for hidden-unit permutation symmetries (Navon et al., 2023; Zhou et al., 2023a), while graph-based metanetworks encode parameterized computations across diverse architectures (Lim et al., 2024a; Kofinas et al., 2024). Related approaches include learned weight alignment (Navon et al., 2024) and prediction of functional similarity from architectural features (Haber & Schneidman, 2022). Building on this perspective, our work draws on graph-based methods for processing trained networks. We use these methods to study a different prediction problem: learning activation-derived relations between internal sites from their parameterized upstream computations, and used for retrieval and stitching candidate selection.

3 Parameter-based Analysis for Relational Inference of Sites

Our goal is to predict representation similarity between internal sites from architecture and trained weights alone without evaluating the compared models on probe inputs at inference. Motivated by the geometry and symmetries of weight space, we represent each site by a graph of its upstream computation, and train a metanetwork to recover similarity scores computed from reference responses.

Notation. Let $\mathcal{F}_{\mathrm{tr}} = \{f_{\boldsymbol{\theta}_m}\}_{m=1}^{M}$ denote the population of models available for meta-training, where each base model $f_{\theta_m} : \mathcal{X} \to \mathcal{Y}_{\mathrm{task}}$ is parameterized by $\boldsymbol{\theta}_m \in \Theta$. Each index m refers to one independently initialized and trained model. Model m contains a set of internal representation sites $\mathcal{S}_m$. We write $u = (m, s)$, with $s \in \mathcal{S}_m$, for an internal representation site in model m, where a *representation site* is a tensor-valued location in the forward computation, e.g., the output of a residual block or the class-token activation after a Transformer block. We write $\mathcal{C}_u$ for the architecture and trained

parameters of the computation upstream of u, which allows us to distinguish two sites with identical local layer parameters whose preceding parameterized computations produce different features.

3.1 Problem formulation

Let $\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^n$ be a shared probe dataset from a fixed reference distribution on $\mathcal{X}$. For a site u, let $\boldsymbol{r}_u(\boldsymbol{x}) \in \mathbb{R}^{p_u}$ denote its representation of input $\boldsymbol{x}$. We apply a fixed extraction rule at each site, such as selecting a token, pooling spatial features, or vectorizing the output. Its reference responses are $\boldsymbol{H}_u = (\boldsymbol{r}_u(\boldsymbol{x}_i)^\top)_{i=1}^n \in \mathbb{R}^{n \times p_u}$, where rows correspond to the same ordered inputs across sites while feature dimensions may differ. We keep the reference distribution, preprocessing, and extraction rules fixed across all comparisons. Each relation type h defines a target distance based on activations $y_{uv}^{(\mathsf{h})} = \delta_{\mathsf{h}}(\boldsymbol{H}_u, \boldsymbol{H}_v)$ for comparing the activations at two sites. In this work, we consider CKA-based dissimilarities (Kornblith et al., 2019; Brockmeier et al., 2017), GULP distance (Boix-Adsera et al., 2022), MutualNN dissimilarity (Huh et al., 2024), or held-out linear-readout disagreement (see App. B for further details). For measures naturally defined as similarities, such as CKA or MutualNN, we use the corresponding dissimilarity so that all targets have the same orientation.

We split the independently trained models into disjoint meta-training, validation, and test populations before constructing site pairs. For every pair of sites (u, v) and every relation type h, we construct the example $(\mathcal{C}_u, \mathcal{C}_v, y_{uv}^{(\mathsf{h})})$. Let $\mathcal{T}_{\text{tr}}$ denote the collection of all such examples constructed from the meta-training population. A metanetwork $g_{\boldsymbol{\omega}}$ parameterized by $\boldsymbol{\omega}$ is trained by minimizing

$$\min_{\boldsymbol{\omega}} \frac{1}{|\mathcal{T}_{\text{tr}}|} \sum_{(\mathcal{C}_u, \mathcal{C}_v, y_{uv}^{(\mathsf{h})}) \in \mathcal{T}_{\text{tr}}} \ell_{\mathsf{h}}(g_{\boldsymbol{\omega}}(\mathcal{C}_u, \mathcal{C}_v, \mathsf{h}), y_{uv}^{(\mathsf{h})}). \tag{1}$$

We define the loss ℓ_h in Sec. 3.3. At inference, we evaluate whether g_ω can predict the representation similarity score for sites from independently trained models that were not seen during meta-training.

3.2 Encoding the computation that produces a representation

We represent each internal representation site u by the parameterized computation upstream of that site. To construct its graph, we build on the parameter-graph representations of neural networks by Lim et al. (2024a); Kofinas et al. (2024). The graph contains the operations and trained parameters that produce the representation at u, with u marked as the target site, given by

$$G_u = \Psi(\mathcal{C}_u) = (V_u, E_u, \boldsymbol{X}_u, \boldsymbol{A}_u, t_u), \tag{2}$$

where Ψ is deterministic, V_u and E_u are the nodes and directed edges, $\boldsymbol{X}_u$ and $\boldsymbol{A}_u$ contain node and edge attributes, and t_u identifies the target site. The graph encoding preserves the structure of the upstream computation and the correspondences between its parameter tensors.

These correspondences are important since neural parameters have permutation symmetries (see Fig. 3). For example, consider two linear maps, hidden coordinate i corresponds to row i of the first weight matrix and column i of the second. Simultaneously permuting these indices gives a different parameterization of the same network function (Hecht-Nielsen, 1990; Ainsworth et al., 2023; Lim et al., 2024b; Navon et al., 2023). In the graph, this transformation only relabels the corresponding nodes. Our graph ties them to the same coordinate node. The same construction is used for ResNet channels, Transformer feed-forward coordinates, and attention heads. Thus, the graph encoder can treat these discrete reparameterizations as graph relabelings.

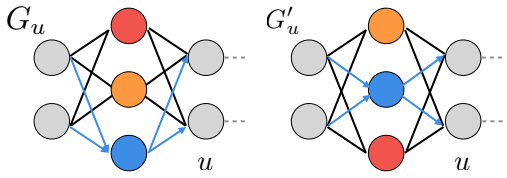

Figure 3: Same function, different parameterization. The graph only changes by a relabeling.

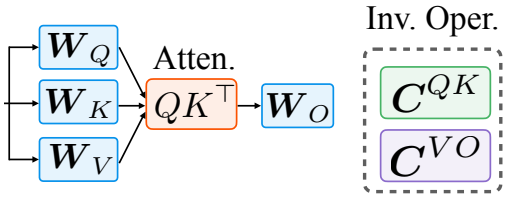

Figure 4: For attention, we encode the composed operator. The graph is invariant to basis changes in query-key and value-output spaces.

Attention additionally admits continuous changes of basis in the query-key and value-output spaces (Vaswani et al., 2017; Theus et al., 2025). Unlike permutation symmetries, these transforma-

tions cannot be represented as graph relabelings, thus they are not handled directly by the graph construction. In our work, we account for these symmetries before graph encoding by considering the corresponding composed query-key and value-output operators. The attention depends on its four factors through the attention circuits via $\boldsymbol{C}^{\mathrm{QK}} = \boldsymbol{W}_Q \boldsymbol{W}_K^\top$ and $\boldsymbol{C}^{\mathrm{VO}} = \boldsymbol{W}_V \boldsymbol{W}_O$, where $\boldsymbol{C}^{\mathrm{QK}}$ determines the bilinear interaction underlying the attention logits and $\boldsymbol{C}^{\mathrm{VO}}$ determines how attended residual-stream features are mapped back into the residual stream (see Fig. 4 for an illustration). We encode their entries as edges associated with a shared head node, preserving the pairing between the QK and VO operators, thus retaining the invariances. The construction for affine self-attention, including query, value, and output biases at residual-stream output sites, is in App. D.3.

3.3 Graph metanetwork estimators

Given two site graphs G_u and G_v, we aim to predict their internal representation dissimilarities. We study two variants that use the same supervision but differ in when the two sites can interact. PARIS-Ind encodes each site independently into a reusable embedding and compares the two embeddings only afterward. PARIS-Joint lets the representations of the two site graphs interact before it predicts their dissimilarities. This comparison tests whether a fixed representation of each site is sufficient or whether pair-dependent processing provides additional information (see App. D for the trade-off).

Independent site embeddings. The independent estimator, PARIS-Ind, maps each site graph to a fixed embedding using a shared graph metanetwork, $\boldsymbol{z}_u = \mathrm{Enc}_\phi(G_u) \in \mathbb{R}^k$, where ϕ denotes the learnable encoder parameters. The encoder first maps the attributes of each node i to an initial hidden state $\boldsymbol{h}_i^{(0)} \in \mathbb{R}^d$. During message-passing round r, an edge $i \to j$ with attributes $\boldsymbol{a}_{ij} \in \mathbb{R}^{d_e}$ produces a message $\boldsymbol{m}_{ij}^{(r)} = \boldsymbol{W}\boldsymbol{h}_i^{(r)} \odot \psi_{\mathrm{edge}}(\boldsymbol{a}_{ij})$, and the receiving node is updated as

$$\boldsymbol{h}_j^{(r+1)} = \mathrm{LN}\left(\boldsymbol{h}_j^{(r)} + \psi\left(\left[\boldsymbol{h}_j^{(r)}, \mathrm{AGG_s}\left((\boldsymbol{m}_{ij}^{(r)})_{i\in\mathcal{N}^-(j)}\right), \mathrm{AGG_m}\left((\boldsymbol{m}_{ij}^{(r)})_{i\in\mathcal{N}^-(j)}\right)\right]\right)\right), \quad (3)$$

where $\psi_{\mathrm{edge}} : \mathbb{R}^{d_e} \to \mathbb{R}^d$ and $\psi : \mathbb{R}^{3d} \to \mathbb{R}^d$ are multilayer perceptrons, $\mathrm{AGG_s}$ and $\mathrm{AGG_m}$ are aggregate functions of the incoming messages using their sum and mean, $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ is learned, $\mathcal{N}^-(j)$ is the incoming-neighbor set of node j, and LN denotes layer normalization.

After R message-passing rounds, the encoder pools the final node states both over the nodes associated with the target representation site and over the complete graph. A readout network maps the resulting summary to $\boldsymbol{z}_u \in \mathbb{R}^k$. Given two sites u and v, relation h is learned from the distance

$$d_{\phi,\mathsf{h}}(u, v) = \|\boldsymbol{W}_{\mathsf{h}}(\boldsymbol{z}_u - \boldsymbol{z}_v)\|_2, \quad (4)$$

that defines a pseudometric on the site-graph representation, where $\boldsymbol{W}_{\mathsf{h}} \in \mathbb{R}^{k_{\mathsf{h}}\times k}$ is a learned relation-specific metric head. Thus, the graph metanetwork assigns each site an embedding that can be reused in comparisons with multiple other sites. A monotone calibration then converts it to the scale of the target $\widehat{y}_{uv}^{(\mathsf{h})} = q_{\mathsf{h}}(d_{\phi,\mathsf{h}}(u, v)) = 1 - \exp(-\varepsilon_{\mathsf{h}}(d_{\phi,\mathsf{h}}(u, v)))$ for $\varepsilon_{\mathsf{h}} > 0$. It is worth noting that Eq. (4) is related to multidimensional scaling through distance fitting, which assigns coordinates to objects so that their pairwise distances approximate observed dissimilarities (Cox & Cox, 2008). In our setting, the objects are internal sites, and the dissimilarities are activation-derived relations.

Joint comparison metanetwork. PARIS-Joint processes a pair of sites together. Given G_u and G_v, it first encodes the node tokens of both graphs and then applies bidirectional cross-attention between them. The model allows information from v to affect how it processes u, and vice versa, before it produces the final scores. It predicts all supervised relation types for that site pair:

$$\widehat{\boldsymbol{y}}_{uv} = \mathrm{Comp}_\vartheta(G_u, G_v) = \left(\widehat{y}_{uv}^{(1)}, \ldots, \widehat{y}_{uv}^{(H)}\right) \in [0, 1]^H, \quad (5)$$

where ϑ is the learnable parameters of the comparator and $\widehat{y}_{uv}^{(\mathsf{h})}$ are the learned dissimilarity associated with the relation $\mathsf{h} \in \mathcal{H}$. We refer *joint* to this pair-dependent processing. Unlike PARIS-Ind, PARIS-Joint does not assign a fixed representation to a site independently of its comparison partner.

Comparing sites across two networks. Both variants predict one score for one pair of internal representation sites at a time. To compare the sites of two networks, we apply the estimator to all cross-network pairs. Let models A and B contain the site sets $\mathcal{S}_A$ and $\mathcal{S}_B$. For relation type h, the target and predicted pairwise dissimilarity matrices are respectively defined by

$$\boldsymbol{D}_{\mathsf{h}}(A, B)_{st} = y_{(A,s),(B,t)}^{(\mathsf{h})}, \qquad \widehat{\boldsymbol{D}}_{\mathsf{h}}(A, B)_{st} = \widehat{y}_{(A,s),(B,t)}^{(\mathsf{h})}, \quad (6)$$

where $s \in \mathcal{S}_A$ and $t \in \mathcal{S}_B$. For PARIS-Ind, the entries of $\widehat{\boldsymbol{D}}_{\mathsf{h}}$ come from the distance in Eq. (4), and for PARIS-Joint, they come directly from the joint comparator in Eq. (5).

Each row of $\boldsymbol{D}_{\mathsf{h}}(A, B)$ ranks the sites of model B relative to one query site in model A. Thus, $\boldsymbol{D}_{\mathsf{h}}(A, B)$ can be used for layer retrieval, stitching-candidate selection, and sequence alignment (Kornblith et al., 2019; Bansal et al., 2021; Nguyen et al., 2023). Transforming this matrix to one scalar "network distance" (Klabunde et al., 2025a) would introduce additional assumptions on how sites should be aggregated or matched, and a separate set-level aggregation needs to be applied (e.g., see App. D.3 for Hausdorff construction (Huttenlocher et al., 1993)). Thus, in this work, we keep site-pair prediction separate from any set-level aggregation, and leave it for future exploration.

Shared supervision and training loss. We train PARIS-Ind and PARIS-Joint on the same meta-training site pairs. For a training pair $p = (u, v)$, let $y_p^{(\mathsf{h})}$ denote the target dissimilarity and $\widehat{y}_p^{(\mathsf{h})}$ the model prediction. We use a smooth regression loss $\mathcal{L}_{\text{reg}}^{(\mathsf{h})} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \text{SmoothL1}(\widehat{y}_p^{(\mathsf{h})}, y_p^{(\mathsf{h})})$ with an ordinal loss that encourages the predicted distances to preserve the ordering of the target relations

$$\mathcal{L}_{\text{rank}}^{(\mathsf{h})} = \mathbb{E}_{a,b \sim \mathcal{P}} \log(1 + \exp[-\tau \ \text{sign}(y_a^{(\mathsf{h})} - y_b^{(\mathsf{h})})(\widehat{y}_a^{(\mathsf{h})} - \widehat{y}_b^{(\mathsf{h})})]). \tag{7}$$

The ranking term has a pairwise logistic loss used in Bradley–Terry form (Bradley & Terry, 1952) and RankNet (Burges et al., 2005), which penalizes the model when the predicted ordering disagrees with the target ordering. For PARIS-Ind, we additionally discourage embedding collapse using $\mathcal{L}_{\text{var}} = \max(0, \eta - \frac{1}{k} \sum_{j=1}^{k} \text{sd}_u(\boldsymbol{z}_{u,j}))$, where sd_u is the empirical standard deviation. We select models using the mean correlation between predicted and target dissimilarities. While we train on pairs of representation sites, this objective is not standard binary contrastive learning (Hadsell et al., 2006; Khosla et al., 2020) as there are no positive-vs-negative pair labels. The regression and ranking objectives train the estimators to recover the magnitude and ordering of representation relations. A comparison with contrastive learning and its extension is provided in App. D.3.

3.4 Theoretical properties

We summarize the theoretical properties of the proposed methods here and provide the formal statements, assumptions, detailed constructions, and proofs in App. C. We study three main questions: *(i)* Which properties follow from how we represent and compare sites? *(ii)* How do independent and joint estimators differ in the scores they can represent? *(iii)* Under what conditions do accurate relation predictions support layer retrieval and stitching-candidate selection?

Both PARIS variants respect the parameter symmetries accounted for by the site-graph construction and are symmetric under exchange of the compared sites. These properties follow from the encoding and comparator design. We refer to App. C.1, where we provide the statement and proofs. We next relate the geometric restrictions imposed by the independent and joint comparison architectures.

Proposition 1 (Reference geometry; informal)**.** *For $N \geq 2$ fixed sites evaluated on a common reference sample, we assume nonzero centered Gram matrices for chordal linear CKA and fixed hard readouts with common class semantics for readout disagreement. Chordal linear CKA and GULP can be expressed as Euclidean distances between reference descriptors assigned separately to each site. Hard-readout disagreement can be expressed as a squared Euclidean distance.*

Proposition 2 (Structural properties; informal)**.** *Eq. (4) is a pseudometric, and $q_{\mathsf{h}} \circ d_{\phi,\mathsf{h}}$ is also a pseudometric and preserves the candidate ranking.* PARIS*-Joint compares each site with its partner. This gives the estimator more flexibility, but it does not enforce the triangle inequality.*

The formal statements and proof of Prop. 1 and Prop. 2 are in App. C.2.

We further analyze when accurate scores help us select useful candidates. For layer retrieval, we bound how much worse the selected candidate can be than the best candidate in the list, measured by reference dissimilarity. The bound depends on the MSE of the predicted scores and on the number of candidates. For stitching, we consider two sources of error separately. First, screening may remove useful candidates. Second, after screening, we may choose the wrong candidate because we evaluate the shortlist on a finite, independent validation sample. If the list keeps a candidate with the smallest predicted dissimilarity, we can bound the error introduced by screening. We refer to App C.3 for further statements and details. These results show that accurate comparison scores can help with screening. While they characterize the consequences of accurate predictions, whether such accuracy transfers to unseen models is evaluated empirically in the next section.

4 EXPERIMENTS

We begin with a qualitative analysis of representation-site correspondence across network depths, which examines the structure of the pairwise relations predicted by our method. We then quantitatively evaluate the performance of PARIS on two representation search tasks: *(i)* representation relation prediction and *(ii)* layer retrieval. We further assess its utility for stitching candidate selection. The experimental protocol and implementation details are provided in App. D. Additional results, including an ablation study and meta-training across diverse architectures, are in App. E.

4.1 REPRESENTATION-SITE CORRESPONDENCE

To illustrate whether our method recovers meaningful structure across networks, we first use a qualitative example to inspect how it compares internal representations across networks of different depths. We examine a cross-depth correspondence map between ResNet56 and ResNet20 (He et al., 2016) to explore which sites in the shallower network are preferred counterparts of sites in the deeper network. We compare models trained on CIFAR-10 and the predictions are produced by PARIS-Ind for HSIC-based CKA (Kornblith et al., 2019). We index the 27 selected ResNet-56 sites and the nine selected ResNet-20 sites in forward computational order. Fig. 5 exhibits a depth-ordered band of stronger predicted correspondence. As the query moves from early to later ResNet56 sites, the region of stronger correspondence moves predominantly toward later ResNet20 sites. It is qualitatively consistent with prior CKA analyses reporting stage-dependent similarity in residual networks and correspondence across network depths (Kornblith et al., 2019; Nguyen et al., 2021). This pattern is approximately stage-wise, *many-to-one* correspondence under the relation, i.e., consecutive sites in the deeper network share similar or neighboring preferred counterparts in the shallower network. Directly computing the reference matrix requires running both base models, storing their responses, and evaluating CKA for every site pair. In contrast, PARIS-Ind encodes each site once and reuses the resulting embedding across comparisons at inference. Their complexity analysis is in App. D.

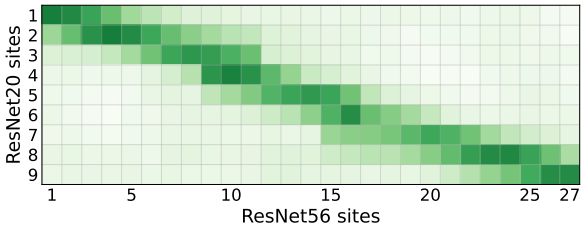

Figure 5: Predicted correspondence between networks of different depths: between 27 ResNet56 sites (columns) and nine ResNet20 sites (rows), ordered from earlier to later sites. Darker green indicates stronger correspondence.

4.2 REPRESENTATION-SCORE PREDICTION

We next evaluate the score prediction, i.e., whether a predictor can estimate the reference dissimilarity between two internal representations from the computations that produce them. Each test example consists of two sites from test models. The predictor receives the sites' upstream architectures and trained parameters, but not their representations on the reference probe inputs. We use the corresponding reference score to evaluate the prediction. We consider five comparison criteria: chordal linear CKA distance, HSIC-based CKA dissimilarity, GULP distance, MutualNN dissimilarity, and disagreement between fitted linear readouts on held-out data. We evaluate ResNet and ViT (Dosovitskiy et al., 2021) populations on CIFAR-10 and CIFAR-100. For each architec-

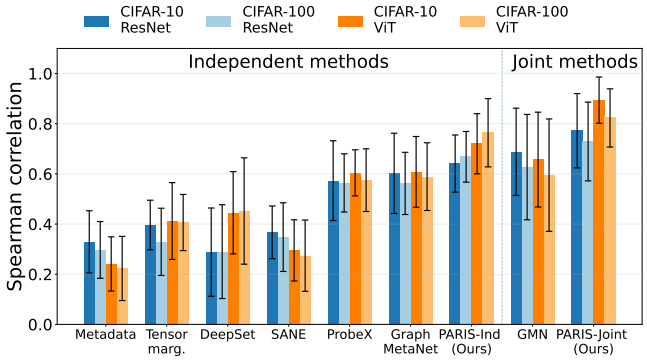

Figure 6: Spearman correlation (↑) between predicted and ground-truth representation relations across ResNet and ViT populations on CIFAR-10 and CIFAR-100.

ture family and downstream task, we construct a population of models with varying widths and depths. Representation sites correspond to tensor-valued locations in each model's forward computation. We compare PARIS against seven baselines: *(i)* architectural metadata, which use upstream computational depth, operator-type counts, target tensor shape, and upstream parameter count, *(ii)* tensor marginal (weight statistics), which summarize the marginal distribution of entries in each upstream parameter tensor (Unterthiner et al., 2020), *(iii)* DeepSets (Zaheer et al., 2017), *(iv)* SANE (Schürholt et al., 2024), *(v)* ProbeX (Horwitz et al., 2025), *(vi)* Graph Metanetworks (Lim et al., 2024a), and *(vii)* Graph Matching Network (GMN) (Li et al., 2019). We adapt the baselines to predict comparison scores from their respective descriptions of the upstream computation. We measure prediction quality using Spearman correlation between predicted and reference dissimilarities across five scores and three seeds. Detailed description of baselines and their adaptations are in App. D.3.

Fig. 6 reports the Spearman correlation between predicted and reference dissimilarities. Among methods that encode representation sites independently, PARIS-Ind obtains the best Spearman correlation in three of the four settings, with noticeably large gains on ViTs. Jointly encoding a pair of representation sites further improves performance. These results indicate that PARIS learns representations that are informative about functional relationships between model internal representations, with the joint variant providing the strongest agreement with ground-truth relation rankings. The two variants, however, have different computational costs at search time (inference). PARIS-Ind computes each site embedding once and reuses it across all comparisons. After this encoding step, searching a large candidate bank only requires evaluating distance heads between stored embeddings. PARIS-Joint instead applies cross-attention separately to each query-candidate pair. Thus, the joint model trades higher score accuracy for more pair-dependent computation, whereas the independent model is better suited to repeated comparisons and large candidate banks.

4.3 Layer retrieval

Next, we investigate the capabilities of PARIS on the layer retrieval task, where a query representation site is used to retrieve related sites from a bank spanning multiple models. This setting tests whether the learned relation predictors can identify close correspondences for individual queries. We use the same model populations, five reference relations, and model splits as in Sec. 4.2. Both queries and candidate sites are drawn from test models. For each reference relation, we rank the candidates by their predicted dissimilarity and return the top five. We compare against the same competing baselines as in Sec. 4.2, and additionally include activation-based similarity methods SVCCA (Raghu et al., 2017) and PWCCA (Morcos et al., 2018). Retrieval quality is measured using mean Regret@5, defined as the difference between the smallest reference dissimilarity among the five retrieved sites and the smallest reference dissimilarity in the full bank, averaged over queries. Lower Regret@5 indicates that the retrieved set contains a candidate closer to the best available match.

Fig. 7 presents the layer retrieval performance. Our method reduces retrieval regret across both architectures and datasets. Among methods that encode representation sites independently, ours achieves the lowest Regret@5. In addition, jointly modeling query-candidate pairs further improves retrieval quality and consistently outperforms GMN. These results show that the relations learned by our approaches transfer effectively from relation prediction to retrieval, enabling it to identify internal representation sites that closely match the best available candidates in a large cross-model bank. Our metanetwork performs well on both relation prediction and layer retrieval, two complementary evaluations that assess global rank agreement across site pairs and the identification of close matches for individual queries, supporting amortized search over model collections.

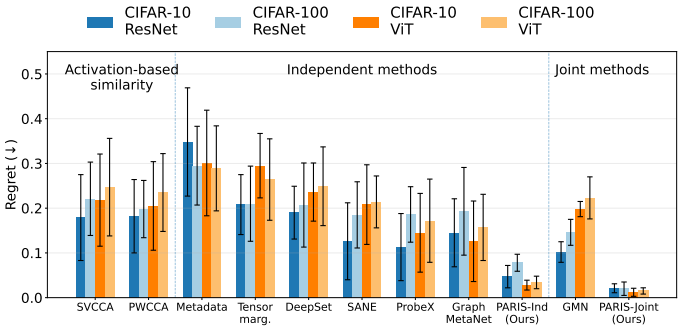

Figure 7: Layer retrieval across model populations, measured by Mean Regret@5 (↓) for retrieving representation sites from a candidate bank spanning multiple test models.

4.4 STITCHING CANDIDATE SELECTION

Finally, we evaluate the effectiveness of PARIS for stitching candidate selection, which identifies connections that preserve downstream performance under an adapter-fitting budget. We adopt the frozen-model stitching protocol of Bansal et al. (2021) and evaluate ResNet-to-ResNet and ViT-to-ViT stitching on CIFAR-10 and CIFAR-100. Each selection fixes a top model from the unseen trained models and provides every selector with the same pool of 20 eligible connections to bottom models from other unseen models. Each candidate connects the initial layers of a bottom model to the later layers of the top model through a trainable adapter (stitching layer). Each selector then chooses five candidates before their adapters are fitted. We consider the same competing baselines as in Sec. 4.2 without training them on stitching outcomes, and additionally include SN-Net-H and KLAS-score, where SN-Net-H adapts SN-Net's heuristics for choosing model pairs and matching blocks (Pan et al., 2023), and KLAS-score ranks candidates using normalized directional KL divergences between the class-probability distributions of fitted linear readouts (Sanyal et al., 2026). Both are evaluated as candidate selectors under the same fitting protocol (see App. D.4 for their detailed adaptations). We report the stitching penalty (Bansal et al., 2021), where the original top model's test accuracy minus the selected stitched model's test accuracy in percentage points.

Fig. 8 reports the performance of the stitching candidate selection methods. We see that our approach achieves the lowest stitching penalty among budgeted selectors. Specifically, PARIS-Ind performs best on CIFAR-10 while PARIS-Joint performs best on CIFAR-100, approaching the performance of full search. This suggests that the learned predictor provides a useful signal for identifying compatible connections before adapter fitting. Importantly, PARIS is not trained on stitching outcomes, suggesting that relational supervision derived from activation patterns transfers effectively to downstream model composition. At the same time, the consistent advantage of PARIS-Joint in relation prediction does not translate into uniformly better stitching selection. The strong performance of PARIS-Ind suggests that the learned site embeddings already capture much of the information needed for effective budgeted selection, without requiring pair-adaptive comparison. In summary, these results support the usefulness of our methods for selecting stitching candidates.

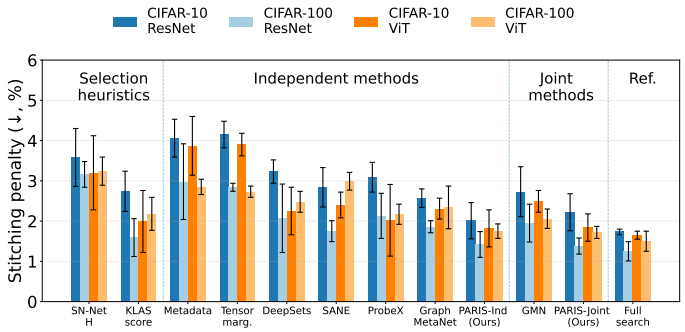

Figure 8: Stitching candidate selection for ResNet-to-ResNet and ViT-to-ViT connections on CIFAR-10 and CIFAR-100. Performance is measured by the stitching penalty (↓).

5 DISCUSSION

In this paper, we studied whether representation relations can be learned from the parameterized computations that produce neural representations. Using activation-derived supervision across independently trained models, our method predicts relations between internal representation sites in unseen models. The results demonstrate that these predictions support layer retrieval and the selection of stitching candidates before adapter fitting. The comparison between the two variants shows that joint processing improves relation prediction and retrieval, while independently computed embeddings can already support effective stitching selection. Beyond their utility for layer retrieval and stitching candidate selection, these findings suggest that rich representational relations are learnable from neural network artifacts across model populations. Some limitations of our study are that graph construction and encoding introduce their own costs as a metanetwork (i.e., amortized method), and any computational advantage depends on model size and the number of comparisons, in addition, the current evaluation is limited to vision tasks. One promising future direction is to extend the framework to multimodal model collections and investigate whether neural network artifacts can be utilized for cross-modal representational alignment. This could provide a complementary parameter perspective on the Platonic Representation Hypothesis (Huh et al., 2024) and on recent reassessments of the extent and robustness of this convergence (Gröger et al., 2026; Koepke et al., 2026).

AI USE STATEMENT

In this work, we used AI tools to assist with writing and editing, including grammar, spelling, phrasing, clarity, style, formatting, and LaTeX presentation. We also used AI tools as coding assistants for figure polishing and formatting. The authors reviewed and tested all AI-suggested code before use. The research direction, main ideas, theoretical claims, and experimental design are original contributions of the authors. The authors reviewed and verified all technical content and take full responsibility for the manuscript.

ETHICS STATEMENT

This work studies representation search by learning to predict similarity between internal neural network representations from model artifacts. We develop graph metanetworks that account for parameter symmetries and compare internal representation sites through the upstream computations that produce them. We support the method with theoretical properties and computational experiments on populations of ResNets and Vision Transformers trained on the public CIFAR-10 and CIFAR-100 datasets. Our experiments focus on comparing representations, retrieving layers, and selecting candidates for model stitching. We use the predicted similarity as a screening signal. Any reuse of model components requires validation for the target task. This study uses only technical model artifacts and public benchmark datasets. It does not involve human participants, personal data, or identifiable information. Thus, we identify no concerns related to privacy, consent, or harm to human subjects, and the study did not require additional ethical approval.

REPRODUCIBILITY STATEMENT

We present the theoretical properties and proofs in App. C. These cover the symmetries of site graphs and predictors, the geometry of independent and joint scores, and the relation between score error, layer retrieval, and stitching. For each result, we present the construction, formal statement, assumptions, and proof. App. D provides our experiments on representation search, representation similarity, and method design. It also includes the full experimental setup and implementation details. We will release the source code upon publication.

REFERENCES

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6430–6439, 2019.

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pp. 384–392, 2019.

Yunsheng Bai, Hao Ding, Ken Gu, Yizhou Sun, and Wei Wang. Learning-based efficient graph similarity computation via multi-scale convolutional set matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 3219–3226, 2020.

András Balogh and Márk Jelasity. On the functional similarity of robust and non-robust neural representations. In *International Conference on Machine Learning*, pp. 1614–1635. PMLR, 2023.

András Balogh and Márk Jelasity. How not to stitch representations to measure similarity: Task loss matching versus direct matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15472–15480, 2025.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in neural information processing systems*, 34:225–236, 2021.

Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1607–1626. PMLR, 2022.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*, 2019.

Yiqing Bo, Ansh Soni, Sudhanshu Srivastava, and Meenakshi Khosla. Evaluating representational similarity measures from the lens of functional correspondence. *arXiv preprint arXiv:2411.14633*, 2024.

Enric Boix-Adsera, Hannah Lawrence, George Stepaniants, and Philippe Rigollet. GULP: a prediction-based metric between representations. *Advances in Neural Information Processing Systems*, 35:7115–7127, 2022.

Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward relu neural network. *Machine learning*, 112(11):4431–4493, 2023.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Austin J Brockmeier, Tingting Mu, Sophia Ananiadou, and John Y Goulermas. Quantifying the informativeness of similarity measurements. *Journal of Machine Learning Research*, 18(76): 1–61, 2017.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.

Irene Cannistraci, Luca Moschella, Marco Fumero, Valentino Maiorca, and Emanuele Rodolà. From bricks to bridges: Product of invariances to enhance latent space communication. In *The Twelfth International Conference on Learning Representations*, 2024.

An Mei Chen, Haw-minn Lu, and Robert Hecht-Nielsen. On the geometry of feedforward neural network error surfaces. *Neural computation*, 5(6):910–927, 1993.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Wei Chen, Zichen Miao, and Qiang Qiu. Inner product-based neural network similarity. *Advances in Neural Information Processing Systems*, 36:73995–74020, 2023.

Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pp. 315–347. Springer, 2008.

Adrián Csiszárik, Péter Kőrösi-Szabó, Akos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of CKA as a similarity measure in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Jörn Diedrichsen and Nikolaus Kriegeskorte. Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLoS computational biology*, 13(4):e1005508, 2017.

Frances Ding, Jean-Stanislas Denain, and Jacob Steinhardt. Grounding representation similarity through statistical testing. *Advances in neural information processing systems*, 34:1556–1568, 2021.

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. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Gabriel Eilertsen, Daniel Jönsson, Timo Ropinski, Jonas Unger, and Anders Ynnerman. Classifying the classifier: dissecting the weight space of neural networks. In *ECAI 2020: 24th European Conference on Artificial Intelligence, 29 August–8 September 2020, Santiago de Compostela, Spain–Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, pp. 1119–1126. SAGE Publications 1 Oliver's Yard, 55 City Road, London, EC1Y 1SP, 2020.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12, 2021.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

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

ML Gonzalez, Melwina Albuquerque, Laurence Wroe, Jacob Meyer Cohen, Logan Riggs Smith, and Thomas Dooms. When are two networks the same? tensor similarity for mechanistic interpretability. *arXiv preprint arXiv:2605.15183*, 2026.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005.

Fabian Gröger, Shuo Wen, and Maria Brbic. Revisiting the platonic representation hypothesis: An aristotelian view. In *Forty-third International Conference on Machine Learning*, 2026.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. Universal neurons in GPT2 language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Adam Haber and Elad Schneidman. Learning the architectural features that predict functional similarity of neural networks. *Physical Review X*, 12(2):021051, 2022.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Sarah E Harvey, David Lipshutz, and Alex H Williams. What representational similarity measures imply about decodable information. In *UniReps: 2nd Edition of the Workshop on Unifying Representations in Neural Models*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pp. 129–135. Elsevier, 1990.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Eliahu Horwitz, Bar Cavia, Jonathan Kahana, and Yedid Hoshen. Learning on model weights using tree experts. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20468–20478. IEEE, 2025.

Jiacong Hu, Jing Gao, Jingwen Ye, Yang Gao, Xingen Wang, Zunlei Feng, and Mingli Song. Model lego: Creating models like disassembling and assembling building blocks. *Advances in Neural Information Processing Systems*, 37:127711–127738, 2024.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*, 2019.

Daniel P Huttenlocher, Gregory A Klanderman, William Rucklidge, et al. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9): 850–863, 1993.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Ioannis Kalogeropoulos, Giorgos Bouritsas, and Yannis Panagakis. Scale equivariant graph metanetworks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

Meenakshi Khosla and Alex H Williams. 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*, pp. 326–341. PMLR, 2024.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *ACM Computing Surveys*, 57(9):1–52, 2025a.

Max Klabunde, Tassilo Wald, Tobias Schumacher, Klaus Maier-Hein, Markus Strohmaier, and Florian Lemmerich. ReSi: A comprehensive benchmark for representational similarity measures. In *The Thirteenth International Conference on Learning Representations*, 2025b.

A Koepke, Daniil Zverev, Shiry Ginosar, and Alexei A Efros. Back into plato's cave: examining cross-modal representational convergence at scale. *arXiv preprint arXiv:2604.18572*, 2026.

Miltiadis Kofinas, Boris Knyazev, Yan Zhang, Yunlu Chen, Gertjan J. Burghouts, Efstratios Gavves, Cees G. M. Snoek, and David W. Zhang. Graph neural networks for learning equivariant representations of neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

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

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 991–999, 2015.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pp. 196–212, Montreal, Canada, 11 Dec 2015. PMLR.

Yujia Li, Chenjie Gu, Thomas Dullien, Oriol Vinyals, and Pushmeet Kohli. Graph matching networks for learning the similarity of graph structured objects. In *International conference on machine learning*, pp. 3835–3845. PmLR, 2019.

Derek Lim, Haggai Maron, Marc T. Law, Jonathan Lorraine, and James Lucas. Graph metanetworks for processing diverse neural architectures. In *The Twelfth International Conference on Learning Representations*, 2024a.

Derek Lim, Theo Putterman, Robin Walters, Haggai Maron, and Stefanie Jegelka. The empirical impact of neural parameter symmetries, or lack thereof. *Advances in Neural Information Processing Systems*, 37:28322–28358, 2024b.

Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. *Advances in Neural Information Processing Systems*, 36:55394–55414, 2023.

Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in neural information processing systems*, 31, 2018.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023.

Aviv Navon, Aviv Shamsian, Idan Achituve, Ethan Fetaya, Gal Chechik, and Haggai Maron. Equivariant architectures for learning in deep weight spaces. In *International Conference on Machine Learning*, pp. 25790–25816. PMLR, 2023.

Aviv Navon, Aviv Shamsian, Ethan Fetaya, Gal Chechik, Nadav Dym, and Haggai Maron. Equivariant deep weight space alignment. In *Forty-first International Conference on Machine Learning*, 2024.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International conference on machine learning*, pp. 7294–7305. PMLR, 2020.

Dang Nguyen, Trang Nguyen, Khai Nguyen, Dinh Phung, Hung Bui, and Nhat Ho. On cross-layer alignment for model fusion of heterogeneous neural networks. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *International Conference on Learning Representations*, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16102–16112. IEEE, 2023.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3962–3971. IEEE, 2019.

Mary Phuong and Christoph H. Lampert. Functional vs. parametric equivalence of ReLU networks. In *International Conference on Learning Representations*, 2020.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in neural information processing systems*, 30, 2017.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Debopam Sanyal, Anantharaman S. Iyer, Alind Khare, Trisha Jain, Akshay Jajoo, Myungjin Lee, James Clayton Kerce, and Alexey Tumanov. KLAS: Using similarity to stitch neural networks for improved accuracy-efficiency tradeoffs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Isaac J Schoenberg. Remarks to maurice frechet's article"sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de hilbert. *Annals of Mathematics*, 36 (3):724–732, 1935.

Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Self-supervised representation learning on neural network weights for model characteristic prediction. *Advances in Neural Information Processing Systems*, 34:16481–16493, 2021.

Konstantin Schürholt, Diyar Taskiran, Boris Knyazev, Xavier Giró-i Nieto, and Damian Borth. Model zoos: A dataset of diverse populations of neural network models. *Advances in Neural Information Processing Systems*, 35:38134–38148, 2022.

Konstantin Schürholt, Michael W. Mahoney, and Damian Borth. Towards scalable and versatile weight space learning. In *Forty-first International Conference on Machine Learning*, 2024.

Konstantin Schürholt, Léo Meynent, Yefan Zhou, Haiquan Lu, Yaoqing Yang, and Damian Borth. A model zoo on phase transitions in neural networks. *arXiv preprint arXiv:2504.18072*, 2025.

Aviv Shamsian, Aviv Navon, David W. Zhang, Yan Zhang, Ethan Fetaya, Gal Chechik, and Haggai Maron. Improved generalization of weight space networks via augmentations. In *Forty-first International Conference on Machine Learning*, 2024.

Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In *International Conference on Machine Learning*, pp. 9722–9732. PMLR, 2021.

Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020.

Damian Smith, Harvey Mannering, and Antonia Marcu. Functional alignment can mislead: Examining model stitching. In *Forty-second International Conference on Machine Learning*, 2025.

Pierre Stock and Rémi Gribonval. An embedding of relu networks and an analysis of their identifiability. *Constructive Approximation*, 57(2):853–899, 2023.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 1199–1208. IEEE, 2018.

Héctor J Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output map. *Neural networks*, 5(4):589–593, 1992.

Norman Tatro, Pin-Yu Chen, Payel Das, Igor Melnyk, Prasanna Sattigeri, and Rongjie Lai. Optimizing mode connectivity via neuron alignment. *Advances in Neural Information Processing Systems*, 33:15300–15311, 2020.

Surat Teerapittayanon, Marcus Comiter, Bradley McDanel, and HT Kung. Stitchnet: Composing neural networks from pre-trained fragments. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pp. 61–68. IEEE, 2023.

Marvin Theiss, Lukas Braun, Andrew M Saxe, and Erin Grant. Parameter symmetries determine representational geometry in overparameterized nonlinear networks. In *ICML 2026 Workshop on Weight-Space Symmetries: from Foundations to Practical Applications*, 2026.

Alexander Theus, Alessandro Cabodi, Sotiris Anagnostidis, Antonio Orvieto, Sidak Pal Singh, and Valentina Boeva. Generalized linear mode connectivity for transformers. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hoang V. Tran, Thieu Vo, An Nguyen The, Tho Tran Huu, Minh-Khoi Nguyen-Nhat, Thanh Tran, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant neural functional networks for transformers. In *The Thirteenth International Conference on Learning Representations*, 2025.

Thomas Unterthiner, Daniel Keysers, Sylvain Gelly, Olivier Bousquet, and Ilya Tolstikhin. Predicting neural network accuracy from weights. *arXiv preprint arXiv:2002.11448*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Verner Vlačić and Helmut Bölcskei. Affine symmetries and neural network identifiability. *Advances in Mathematics*, 376:107485, 2021.

Thieu Vo, Hoang V. Tran, Tho Tran Huu, An Nguyen The, Thanh Tran, Minh-Khoi Nguyen-Nhat, Duy-Tung Pham, and Tan Minh Nguyen. Equivariant polynomial functional networks. In *Forty-second International Conference on Machine Learning*, 2025.

Christopher Wang, Vighnesh Subramaniam, Dan Gutfreund, Boris Katz, Phillip Isola, and Brian Cheung. Model merging via averaged representational similarity. In *ICML 2026 Workshop on Weight-Space Symmetries: from Foundations to Practical Applications*, 2026.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

Jiaqi Wang, Xingyi Yang, Suhan Cui, Liwei Che, Lingjuan Lyu, Dongkuan DK Xu, and Fenglong Ma. Towards personalized federated learning via heterogeneous model reassembly. *Advances in Neural Information Processing Systems*, 36:29515–29531, 2023.

Alex H Williams, Erin Kunz, Simon Kornblith, and Scott Linderman. Generalized shape metrics on neural representations. *Advances in neural information processing systems*, 34:4738–4750, 2021.

Xiaojun Xu, Chang Liu, Qian Feng, Heng Yin, Le Song, and Dawn Song. Neural network-based graph embedding for cross-platform binary code similarity detection. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 363–376, 2017.

Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *Advances in neural information processing systems*, 35:25739–25753, 2022.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International conference on machine learning*, pp. 12133–12143. Pmlr, 2021.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pp. 7252–7261. PMLR, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Binchi Zhang, Zaiyi Zheng, Zhengzhang Chen, and Jundong Li. Beyond the permutation symmetry of transformers: The role of rotation for model fusion. In *Forty-second International Conference on Machine Learning*, 2025.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model spider: Learning to rank pre-trained models efficiently. *Advances in Neural Information Processing Systems*, 36:13692–13719, 2023.

Bo Zhao, Robin Walters, and Rose Yu. Symmetry in neural network parameter spaces. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. *Advances in neural information processing systems*, 36:24966–24992, 2023a.

Allan Zhou, Kaien Yang, Yiding Jiang, Kaylee Burns, Winnie Xu, Samuel Sokota, J Zico Kolter, and Chelsea Finn. Neural functional transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

APPENDICES

A Extended related work **19**

B Representation (dis)similarity measures **23**

C Theoretical properties **26**

C.1 Symmetries of the site graphs and predictors 26

C.2 Geometry of independent and joint scores 28

C.3 Score error, layer retrieval and stitching 36

D Experimental setup and details **41**

D.1 Base-model populations and training 41

D.2 Representation sites 42

D.3 Method construction 42

D.4 Baseline adaptations 50

E Additional experiments **54**

E.1 Assessing cross-graph interaction 54

E.2 Ablating regression and ranking supervision 54

E.3 Sensitivity to the sample size 55

E.4 Per-representation score correlation 55

E.5 Meta-training across diverse architectures 55

E.6 Cross-task prediction with partially overlapping label spaces 56

A Extended related work

We expand on the two directions discussed in the main paper and their connections to model composition, parameter symmetries, graph comparison, and pretrained-model selection. Throughout, we distinguish defining a representation relation, learning to predict that relation, and using a predicted correspondence for a downstream intervention.

Representation similarity. Comparing responses to shared inputs provides a way to study learned structure without assuming that individual coordinates correspond. Representational similarity analysis compares patterns of dissimilarity among stimulus responses (Kriegeskorte et al., 2008). In artificial networks, convergent learning studies examine which features and subspaces recur across independent training runs (Li et al., 2015), while subsequent work investigates recurring neurons in language models (Gurnee et al., 2024) and broader representational convergence across models and modalities (Huh et al., 2024). Analyses of width and depth further show that architectural changes affect the organization of learned representations (Nguyen et al., 2021). These studies motivate searching for correspondences beyond identically indexed layers or neurons, while leaving the appropriate notion of correspondence dependent on the scientific question.

Different comparison methods emphasize different properties of the responses. SVCCA and PWCCA characterize relationships between activation subspaces (Raghu et al., 2017; Morcos et al., 2018), whereas CKA compares centered similarity matrices over examples (Kornblith et al., 2019). Generalized shape metrics formalize representation comparison under specified transformation groups (Williams et al., 2021), and soft matching preserves information about individual-neuron tuning while accommodating different numbers of units (Khosla & Williams, 2024). Neighborhood overlap supplies a more local criterion, as in the mutual-nearest-neighbor comparisons used by Huh et al. (2024). These approaches do not need to induce the same ranking of candidate sites: agreement in global geometry, local neighborhoods, and individual-unit responses constitutes different evidence of shared structure. Balogh & Jelasity (2023) show that robust and non-robust representations can be compatible in ordinary classification accuracy while differing in robustness compatibility, with the relationship changing across layers.

Predictive criteria offer a complementary perspective. GULP defines a family of distances motivated by regularized linear prediction (Boix-Adsera et al., 2022). Geometric and predictive interpretations are not mutually exclusive: Harvey et al. (2024) relate several similarity measures to the alignment of optimal linear readouts across a distribution of decoding tasks. Fitted probes instead evaluate information accessible under a particular readout family and training procedure (Alain & Bengio, 2016; Hewitt & Liang, 2019). Comparing the resulting predictions provides a task-specific behavioral relation, whose interpretation depends on the readout protocol and the chosen agreement statistic (Bo et al., 2024). Statistical evaluations, analyses of CKA reliability, and the ReSi benchmark demonstrates the importance of specifying what a similarity measure is intended to capture (Ding et al., 2021; Davari et al., 2023; Klabunde et al., 2025b).

Parameter-based representation comparison also has direct precedents. Chen et al. (2023) introduce a filter-subspace similarity for convolutional networks whose filters are decomposed into model-specific atoms and coefficients shared across networks. This construction permits comparison from filter atoms without probing forward passes, with theoretical and empirical connections to activation-based similarities. Their applications also include retrieval of related models for knowledge transfer. Thus, neither avoiding probing activations nor searching a model collection alone distinguishes our approach. PARIS instead learns selected activation-defined relations between internal sites from their upstream computations, without imposing this shared-coefficient construction on the base models. It targets retrieval in held-out models from the studied populations.

A different analytical approach is developed by Gonzalez et al. (2026), who define tensor similarity for polynomial and multilinear network architectures. Their recursive tensor comparison accounts for cross-layer computation and parameter symmetries without sampling input responses. Localized variants compare output components and functional differences. Under their stated assumptions, a normalized similarity of one characterizes functional equivalence up to a positive scalar. This differs from our learning objective: PARIS predicts empirical activation relations for the base model architectures and we do not aim for providing an algebraic certificate of global functional equivalence.

PARIS is complementary to these comparison criteria. We use measured activation-derived relations as supervision and learn to predict them from parameterized upstream computations. Direct evaluation of these activation-defined targets requires obtaining the relevant responses or derived statistics, although these can be cached. Our search-time predictor does not collect new base-model activations. Its predictions remain tied to the reference-input distribution and, where applicable, the readout protocol used to define the target. Thus, fidelity to the chosen relation and the usefulness of that relation for interpretation or reuse are separate questions.

Model stitching, latent alignment, and component reuse. Feature-transfer studies show that reusing network components depends on both feature specialization and interactions between neighboring layers (Yosinski et al., 2014). Representation-equivalence experiments study transformations that allow one representation to substitute for another (Lenc & Vedaldi, 2015). Model stitching develops this perspective by connecting the lower portion of one trained network to the upper portion of another and evaluating the resulting computation (Bansal et al., 2021; Csiszárik et al., 2021). Stitching supplies a task-dependent test of compatibility, whereas an activation-based statistic compares the representations under a specified geometric or predictive criterion. These objectives can be related without being identical.

Latent-space communication methods address the coordinate mismatch between independently learned components. Relative representations describe examples through their relationships to reference samples (Moschella et al., 2023), latent-space translation learns explicit alignments between separately trained spaces (Maiorca et al., 2023), and products of invariances combine complementary descriptions to improve communication (Cannistraci et al., 2024). At the level of model collections, Deep Model Reassembly partitions pretrained models and selects blocks for a new network (Yang et al., 2022). StitchNet uses CKA to guide the selection of pretrained fragments (Teerapittayanon et al., 2023), while Stitchable Neural Networks vary connections between pretrained anchors to obtain different accuracy–efficiency trade-offs (Pan et al., 2023). These methods provide concrete motivations for locating useful components within and across trained models.

More recently, KLAS uses KL divergence between the predictive distributions of linear probes attached to intermediate blocks, together with a normalized stitch score, to select candidate connections across pretrained models (Sanyal et al., 2026). This is a direct precedent for using a readout-based relation to guide stitch selection. Its scoring relies on responses to data, whereas PARIS predicts site relations from parameterized computations at search time. Our meta-training targets are representation relations and *not* the accuracy of assembled networks. Any subsequent connector training and stitching validation remain separate from relation prediction.

Our objective is to retrieve candidate sites under a selected representation relation, and we do not directly optimize an assembled network or fit a transformation between latent spaces. PARIS can provide a candidate-identification stage for a subsequent composition procedure, but a predicted match does not itself establish interchangeability. A retrieved component may still require an adapter, fine-tuning, and task-specific validation. Conversely, successful stitching does not establish that two original representations contain the same information: studies of stitching protocols demonstrate that functional alignment can be achieved despite substantial representational differences (Balogh & Jelasity, 2025; Smith et al., 2025). We consequently distinguish representation retrieval from both the construction and validation of a combined model.

Connection to representation-space model merging. Wang et al. (2026) average sample-level representation kernels from independently trained models and optimize a student toward the averaged kernel using differentiable mutual-nearest-neighbor similarity. This suggests a possible extension in which predicted site relations help screen source models or select compatible sites before representation-space merging. Such screening would require validating which relations identify useful sources. Moreover, scalar pairwise relation predictions do not specify the sample-level Gram matrices needed for kernel averaging. The current PARIS objective is not an activation-free replacement for this merging procedure. A fuller integration could use PARIS for candidate selection while retaining response-based merging, or investigate a new prediction target carrying the additional kernel information. These are prospective extensions, not components of the present framework.

Parameter symmetries and structured computation encodings. Neural-network parameters are not unique descriptions of network functions. Consistent hidden-unit relabelings produce equivalent parameterizations, a property studied in early analyses of weight spaces (Hecht-Nielsen, 1990).

Permutation alignment is central to methods for comparing trained solutions and merging models, including neuron-aligned mode connectivity, Git Re-Basin, and matched averaging (Tatro et al., 2020; Ainsworth et al., 2023; Wang et al., 2020). Optimal-transport fusion further permits soft correspondences between models with different widths (Singh & Jaggi, 2020). These methods distinguish parameter coordinates from learned computation. Yet, Soft matching across different architectures does not have to be conflated with an exact function-preserving permutation.

An alternative to explicitly aligning each pair is to incorporate the symmetry into the parameter-processing architecture. DWSNets and neural functional networks respect the coupled action of hidden-unit permutations on neighboring weight tensors (Navon et al., 2023; Zhou et al., 2023a). Neural Functional Transformers and equivariant polynomial architectures extend this approach with more expressive parameter interactions (Zhou et al., 2023b; Vo et al., 2025). Graph Metanetworks and neural-graph representations instead encode parameters together with connectivity, supporting diverse input-network architectures (Lim et al., 2024a; Kofinas et al., 2024). Our site encoding builds on this structured perspective: it describes the entire computation upstream of a selected tensor, not only the parameters of the immediately preceding layer. It also includes a fixed inference state. For example, BatchNorm inference depends on population statistics in addition to trainable parameters (Ioffe & Szegedy, 2015).

Attention introduces internal coordinate choices beyond hidden-unit permutations. The transformer circuits perspective separates the query-key and output-value computations (Elhage et al., 2021), while Transformer-NFN formalizes internal basis symmetries and uses composed query-key and value-output operators (Tran et al., 2025). Related work exploits attention rotation symmetries for model fusion (Zhang et al., 2025). Theus et al. (2025) explicitly uses the composed QK and OV operators as canonical head representations in their Transformer-alignment framework, removing the corresponding internal invertible reparameterizations from the comparison. PARIS builds on this existing reduction into its site-graph encoding. Thus, it avoids distinguishing factorizations that differ only by admissible internal basis changes. This invariance concerns the supported attention computation, not arbitrary transformations of every Transformer implementation. Positional operations and other intervening transformations are respected when determining the admissible symmetry.

The invariances of a parameter encoder are also required to agree with the relation being predicted. Studies of network symmetries connect function-preserving parameter transformations to changes in internal representations (Godfrey et al., 2022), and ScaleGMNs incorporate additional activation-dependent scaling symmetries (Kalogeropoulos et al., 2024). Nevertheless, preserving the end-to-end function does not guarantee that an internal representation retains its geometry. In particular, linear CKA is invariant to coordinate permutations and orthogonal transformations, but not arbitrary anisotropic rescaling (Kornblith et al., 2019). For PARIS, symmetry handling removes irrelevant coordinate choices while retaining distinctions that can affect the target site relation. It is not a claim of invariance to all functionally equivalent parameterizations.

The analysis of Theiss et al. (2026) further sharpens this distinction. They characterize how parameter symmetries reshape representations through addition, duplication, and scaling, and derive conditions under which functionally equivalent nonlinear networks can have arbitrarily dissimilar representations. For PARIS, this implies that invariance needs to be checked against the relation at the queried site, not inferred solely from equality of the network's final output. If an encoding identifies two upstream computations whose target relations to a third site differ, no deterministic predictor on that encoding can reproduce both targets. Accordingly, we seek invariance to target-preserving relabelings and admissible internal basis changes, while retaining reparameterizations that alter the selected representation geometry.

Independent embeddings and interaction-based graph comparison. Learning to compare graph-structured objects provides a direct architectural precedent for our two variants. Graph Matching Networks contrast independently computed graph embeddings with pair-conditioned encodings that exchange information through cross-graph attention (Li et al., 2019). SimGNN combines graph-level descriptors with fine-grained node comparisons (Bai et al., 2019), while GraphSim compares node embeddings at multiple scales (Bai et al., 2020). Graph embeddings have also been used to retrieve related compiled functions from control-flow graphs (Xu et al., 2017). These approaches establish independent encoding and cross-graph interaction as existing strategies for comparing structured objects. Our task instead uses parameterized neural computations as inputs and activation-derived site relations as supervision.

An analogous design space appears in information retrieval. Sentence-BERT and dense passage retrieval use separately computed query and candidate representations (Reimers & Gurevych, 2019; Karpukhin et al., 2020). Poly-encoders study intermediate designs between independent encoding and full pairwise interaction (Humeau et al., 2019), while ColBERT precomputes candidate representations and retains query-dependent late interaction (Khattab & Zaharia, 2020). Thus, reusable candidate features and pair-adaptive scoring are not mutually exclusive. Learned relation modules also provide an alternative to prescribing a fixed distance between embeddings (Sung et al., 2018).

PARIS-Ind and PARIS-Joint instantiate two points in this design space. The independent variant computes a reusable descriptor for each upstream-computation graph and applies relation-specific heads. The joint variant permits interaction between site graphs before producing a score, so its comparison can depend on the particular query-candidate pair. These variants adapt established graph-comparison designs to parameter-based site-relation prediction and allow us to study their trade-offs in this setting. Reusable embeddings avoid repeating candidate encoding across queries. Efficient nearest-neighbor indexing additionally depends on the form of the comparison head. Note that greater pairwise interaction does not by itself guarantee better retrieval: the relevant trade-off is empirical and includes prediction quality, encoding reuse, and per-comparison cost.

Relational knowledge distillation. Our setup is also related to relational knowledge distillation (Park et al., 2019), where a student learns relations from a teacher. We adopt a related principle here, however, the targets compare internal sites from independently trained models, while the predictor receives only the parameterized computations that produce those sites. The goal is to predict the same comparisons for previously unseen models.

Broader weight space learning and pretrained-model selection. Learning over populations of trained networks has been used to infer model characteristics directly from parameters. Early examples predict accuracy from weights (Unterthiner et al., 2020) and identify properties of models and their training configurations (Eilertsen et al., 2020). Hyper-representations learn reusable weight space embeddings for model-characteristic prediction (Schürholt et al., 2021), and SANE extends weight space representation learning through sequential processing of weight subsets, including layer-wise embeddings (Schürholt et al., 2024). Model-zoo datasets and augmentation strategies provide complementary resources for learning and generalization across network populations (Schürholt et al., 2022; Shamsian et al., 2024). Accordingly, neither learning from trained parameters nor producing layer-level descriptors alone distinguishes our setting.

Schürholt et al. (2025) provide model zoos organized by loss-landscape phases, with controlled variation in architecture, size, and data. This is a complementary resource for the population-level learning setting of PARIS. A prospective evaluation could use these populations to distinguish within-phase interpolation from cross-phase generalization, and assess whether relation prediction transfers beyond changes of random seed. Such an evaluation needs to separate complete training runs across meta-training and testing, keeping related checkpoints and their sites in the same split. Phase information can support stratified evaluation without becoming an activation-dependent input to the search-time predictor. Using these datasets and performing these stronger generalization tests are possible extensions, not established results of the present study.

Relational precedents require a more specific comparison. Deep-Align learns a symmetry-aware mapping from network pairs to parameter alignments (Navon et al., 2024), where its objective is to recover permutations. Haber & Schneidman (2022) learn a metric from synaptic differences that predicts response similarity in novel simulated spiking networks, and identify a sparse set of architectural features that supports this prediction. This is an especially close precedent for learning functional relations from network parameters. Yet, PARIS studies a different setting: internal sites of trained deep networks, described by their parameterized upstream computations and compared under specified activation-derived criteria. The resulting relational predictions are used for search across models that were not used for meta-training.

Pretrained-model selection shares the goal of reducing candidate evaluation, but typically ranks models for a target task rather than for a query representation. Task2Vec uses task embeddings to support feature-extractor selection (Achille et al., 2019). LEEP and LogME assess transferability using model predictions or extracted features on target data (Nguyen et al., 2020; You et al., 2021), whereas Model Spider learns model-task compatibility from approximate performance measurements (Zhang et al., 2023). These methods motivate learned surrogates and efficient search over

model collections, but model-task suitability is distinct from a relation between two internal representation sites.

PARIS brings these directions together by treating activation-derived site-pair measurements as supervision for a graph metanetwork. The query is a representation site, the candidates are sites in other trained models, and the predicted quantity is the selected representation relation. At inference, the base-model computation is supplied through its architecture, trained parameters, and fixed inference state. The framework amortizes activation-based relation estimation across a model population, while relation-prediction accuracy and generalization to unseen models remain empirical questions.

B REPRESENTATION (DIS)SIMILARITY MEASURES

This section defines the activation-derived representation relations used in our paper.

Reference observations and notation. Let $\boldsymbol{X} = (\boldsymbol{x}_i)_{i=1}^n$ denote the ordered reference sample, where $n = 1000$ in our experiments. As defined in App. D.2, a representation site u produces the feature vector $r_u(\boldsymbol{x}_i) \in \mathbb{R}^{p_u}$ for reference input $\boldsymbol{x}_i$. The corresponding activation matrix is

$$\boldsymbol{H}_u(\boldsymbol{X}) = \begin{bmatrix} r_u(\boldsymbol{x}_1)^\top \\ \vdots \\ r_u(\boldsymbol{x}_n)^\top \end{bmatrix} \in \mathbb{R}^{n \times p_u}. \tag{8}$$

Rows correspond to the same reference examples in the same order for every site, whereas the feature dimension p_u may differ across sites. For a relation type h, the reference target is written as

$$y_{uv}^{(\mathsf{h})} = \delta_{\mathsf{h}}\left(\boldsymbol{H}_u(\boldsymbol{X}), \boldsymbol{H}_v(\boldsymbol{X})\right), \tag{9}$$

where δ_{h} denotes the relation-specific dissimilarity defined below. Reference-example identities and their ordering are fixed before computing any relation. We do not match examples according to model predictions or reorder the observations separately for different sites or site pairs.

Centering. For relations that require centering over reference examples, define the centering matrix $\boldsymbol{J}_n = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, and the centered activation matrix $\overline{\boldsymbol{H}}_u = \boldsymbol{J}_n\boldsymbol{H}_u$. Left multiplication by $\boldsymbol{J}_n$ subtracts the empirical mean of each feature over the reference examples. It does not center the feature coordinates within an individual example.

Chordal linear-CKA distance. Linear centered kernel alignment (CKA) compares the centered inner-product geometry of two representations (Kornblith et al., 2019; Brockmeier et al., 2017). For brevity, write $\boldsymbol{H}_u = \boldsymbol{H}_u(\boldsymbol{X})$ for the activation matrix of site u. We define its linear Gram matrix and centered Gram matrix as

$$\boldsymbol{K}_u = \boldsymbol{H}_u\boldsymbol{H}_u^\top, \qquad \overline{\boldsymbol{K}}_u = \boldsymbol{J}_n\boldsymbol{K}_u\boldsymbol{J}_n = \overline{\boldsymbol{H}}_u\overline{\boldsymbol{H}}_u^\top, \tag{10}$$

where $\overline{\boldsymbol{H}}_u = \boldsymbol{J}_n\boldsymbol{H}_u$ and $\boldsymbol{J}_n$ is the centering matrix defined above. For sites u and v with nonzero centered Gram matrices, the linear CKA similarity is

$$s_{\mathrm{CKA}}^{\mathrm{lin}}(u,v) = \frac{\langle \overline{\boldsymbol{K}}_u, \overline{\boldsymbol{K}}_v \rangle_F}{\|\overline{\boldsymbol{K}}_u\|_F \|\overline{\boldsymbol{K}}_v\|_F}, \tag{11}$$

where $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_F = \mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{B})$ denotes the Frobenius inner product. Equivalently, the same quantity can be computed directly in feature space as

$$s_{\mathrm{CKA}}^{\mathrm{lin}}(u,v) = \frac{\left\|\overline{\boldsymbol{H}}_u^\top\overline{\boldsymbol{H}}_v\right\|_F^2}{\left\|\overline{\boldsymbol{H}}_u^\top\overline{\boldsymbol{H}}_u\right\|_F \left\|\overline{\boldsymbol{H}}_v^\top\overline{\boldsymbol{H}}_v\right\|_F}. \tag{12}$$

To obtain the dissimilarity, consider the normalized centered Gram descriptor

$$\boldsymbol{B}_u = \frac{\overline{\boldsymbol{K}}_u}{\|\overline{\boldsymbol{K}}_u\|_F}. \tag{13}$$

We then use the unit-range chordal linear-CKA distance

$$\delta_{\mathrm{C}}(u,v) = \frac{1}{\sqrt{2}}\|\boldsymbol{B}_u - \boldsymbol{B}_v\|_F = \sqrt{1 - s_{\mathrm{CKA}}^{\mathrm{lin}}(u,v)}. \tag{14}$$

Because the normalized centered linear Gram matrices have unit Frobenius norm and $s^{\mathrm{lin}}_{\mathrm{CKA}}(u,v) \in [0,1]$ in exact arithmetic, $\delta_{\mathrm{C}}(u,v) \in [0,1]$. The factor $1/\sqrt{2}$ then converts the usual chordal distance $\sqrt{2-2s^{\mathrm{lin}}_{\mathrm{CKA}}(u,v)}$ to the unit interval. We retain the square root in Eq. (14). Hence, $1-s^{\mathrm{lin}}_{\mathrm{CKA}}(u,v)$ is the squared unit-range chordal distance and is not used as an interchangeable target. No additional normalization or rescaling is applied to δ_{C}.

HSIC-based CKA dissimilarity. CKA normalizes the Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2005) to compare the kernel-induced similarity structure of matched reference observations (Kornblith et al., 2019). For site u, the RBF kernel is given by

$$\left(\boldsymbol{K}_u^{\mathrm{rbf}}\right)_{ij} = \exp\left(-\frac{\|r_u(\boldsymbol{x}_i) - r_u(\boldsymbol{x}_j)\|_2^2}{2\sigma_u^2}\right), \tag{15}$$

where $r_u(\boldsymbol{x}_i)$ is the extracted representation vector, and $\sigma_u > 0$ is the bandwidth. The bandwidth determines the scale at which pairwise distances between reference representations are compared.

Let $\boldsymbol{J}_n = \boldsymbol{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ denote the centering matrix. For symmetric kernel matrices $\boldsymbol{K}$ and $\boldsymbol{L}$, the biased empirical HSIC estimator is

$$\mathcal{H}_{\mathrm{b}}(\boldsymbol{K},\boldsymbol{L}) = \frac{1}{(n-1)^2}\operatorname{tr}\left(\boldsymbol{K}\boldsymbol{J}_n\boldsymbol{L}\boldsymbol{J}_n\right). \tag{16}$$

Equivalently, $\mathcal{H}_{\mathrm{b}}(\boldsymbol{K},\boldsymbol{L})$ is proportional to the Frobenius inner product between the centered kernel matrices $\boldsymbol{J}_n\boldsymbol{K}\boldsymbol{J}_n$ and $\boldsymbol{J}_n\boldsymbol{L}\boldsymbol{J}_n$. The HSIC-based RBF-CKA similarity is defined as

$$s^{\mathrm{rbf},\nu}_{\mathrm{CKA}}(u,v) = \frac{\mathcal{H}_{\mathrm{b}}\left(\boldsymbol{K}_u^{\mathrm{rbf}},\boldsymbol{K}_v^{\mathrm{rbf}}\right)}{\sqrt{\mathcal{H}_{\mathrm{b}}\left(\boldsymbol{K}_u^{\mathrm{rbf}},\boldsymbol{K}_u^{\mathrm{rbf}}\right)\mathcal{H}_{\mathrm{b}}\left(\boldsymbol{K}_v^{\mathrm{rbf}},\boldsymbol{K}_v^{\mathrm{rbf}}\right)}}. \tag{17}$$

The supervised dissimilarity is then $\delta_{\mathrm{HSIC,rbf}}(u,v) = 1 - s^{\mathrm{rbf},\nu}_{\mathrm{CKA}}(u,v)$.

GULP distance GULP compares representations through the behavior of regularized linear predictors fitted to their features (Boix-Adsera et al., 2022). We first apply a sitewise scalar normalization,

$$a_u = \left(\frac{1}{n}\|\overline{\boldsymbol{H}}_u\|_F^2\right)^{1/2}, \qquad \boldsymbol{Z}_u = \frac{\overline{\boldsymbol{H}}_u}{a_u}. \tag{18}$$

Define the empirical covariance and cross-covariance matrices

$$\boldsymbol{S}_u = \frac{1}{n}\boldsymbol{Z}_u^\top\boldsymbol{Z}_u, \qquad \boldsymbol{S}_{uv} = \frac{1}{n}\boldsymbol{Z}_u^\top\boldsymbol{Z}_v. \tag{19}$$

For regularization parameter $\lambda_{\mathrm{G}} > 0$, let $\boldsymbol{A}_u = (\boldsymbol{S}_u + \lambda_{\mathrm{G}}\boldsymbol{I}_{p_u})^{-1}$. The squared GULP distance between sites u and v is then give by

$$d_{\mathrm{G},\lambda_{\mathrm{G}}}(u,v)^2 = \operatorname{tr}\left(\boldsymbol{A}_u\boldsymbol{S}_u\boldsymbol{A}_u\boldsymbol{S}_u\right) + \operatorname{tr}\left(\boldsymbol{A}_v\boldsymbol{S}_v\boldsymbol{A}_v\boldsymbol{S}_v\right) - 2\operatorname{tr}\left(\boldsymbol{A}_u\boldsymbol{S}_{uv}\boldsymbol{A}_v\boldsymbol{S}_{uv}^\top\right). \tag{20}$$

We use $\lambda_{\mathrm{G}} = 1$ with covariance matrices normalized by $1/n$ as in Eq. (19). Under the equivalent formulation using the unnormalized matrix $\boldsymbol{Z}_u^\top\boldsymbol{Z}_u$, the corresponding regularizer is $n\lambda_{\mathrm{G}}\boldsymbol{I}_{p_u}$. Thus,

$$\boldsymbol{S}_u + \lambda_{\mathrm{G}}\boldsymbol{I}_{p_u} = \frac{1}{n}\left(\boldsymbol{Z}_u^\top\boldsymbol{Z}_u + n\lambda_{\mathrm{G}}\boldsymbol{I}_{p_u}\right). \tag{21}$$

The same distance can be expressed using the regularized sample-space operator

$$\boldsymbol{P}_u = \frac{1}{n}\boldsymbol{Z}_u\boldsymbol{A}_u\boldsymbol{Z}_u^\top = \boldsymbol{Z}_u\left(\boldsymbol{Z}_u^\top\boldsymbol{Z}_u + n\lambda_{\mathrm{G}}\boldsymbol{I}_{p_u}\right)^{-1}\boldsymbol{Z}_u^\top. \tag{22}$$

With this notation, we have

$$d_{\mathrm{G},\lambda_{\mathrm{G}}}(u,v) = \|\boldsymbol{P}_u - \boldsymbol{P}_v\|_F. \tag{23}$$

For completeness, consider a thin singular-value decomposition

$$\boldsymbol{Z}_u = \boldsymbol{U}_u\boldsymbol{D}_u\boldsymbol{V}_u^\top, \tag{24}$$

where the diagonal entries $s_{u,j}$ of $\boldsymbol{D}_u$ are the singular values. Then

$$\boldsymbol{P}_u = \boldsymbol{U}_u\operatorname{diag}\left(\frac{s_{u,j}^2}{s_{u,j}^2 + n\lambda_{\mathrm{G}}}\right)\boldsymbol{U}_u^\top. \tag{25}$$

Mutual nearest-neighbor dissimilarity. MutualNN compares the local neighborhood structure induced by two representations on the same reference examples (Huh et al., 2024). For each site u, let $\widetilde{r}_u(\boldsymbol{x}_i) \in \mathbb{R}^{p_u}$ denote the feature vector.

For reference example i, let

$$\mathcal{N}_u^{k_{\mathrm{NN}}}(i) = \underset{\substack{k_{\mathrm{NN}} \\ j \in \{1,\dots,n\}\setminus\{i\}}}{\arg\operatorname{top}} \left\{-d_{\mathrm{NN}}\left(\widetilde{r}_u(\boldsymbol{x}_i), \widetilde{r}_u(\boldsymbol{x}_j)\right)\right\} \tag{26}$$

denote the indices of the k_{NN} nearest reference examples to example i at site u. For a pair of sites u and v, MutualNN measures the fraction of neighbors shared by the two representation spaces for each corresponding reference example. We define

$$s_{\mathrm{NN}}(u,v) = \frac{1}{n k_{\mathrm{NN}}} \sum_{i=1}^{n} \left|\mathcal{N}_u^{k_{\mathrm{NN}}}(i) \cap \mathcal{N}_v^{k_{\mathrm{NN}}}(i)\right|. \tag{27}$$

The intersection is taken over reference-example indices. The corresponding dissimilarity target is $\delta_{\mathrm{NN}}(u,v) = 1 - s_{\mathrm{NN}}(u,v)$.

Centered kernel nearest-neighbor alignment. Centered kernel nearest-neighbor alignment (CKNNA) combines kernel-based representation comparison with a restriction to neighborhoods shared between the two representation spaces (Huh et al., 2024). For each site, neighborhoods are constructed as in MutualNN. For a pair of sites u and v, consider the pair-specific shared-neighborhood mask

$$(\boldsymbol{M}_{uv})_{ij} = \mathbb{I}\left[j \in \mathcal{N}_u^{k_{\mathrm{CKNNA}}}(i) \cap \mathcal{N}_v^{k_{\mathrm{CKNNA}}}(i)\right]. \tag{28}$$

Thus, $(\boldsymbol{M}_{uv})_{ij} = 1$ only when reference example j is among the k_{CKNNA} nearest neighbors of reference example i in both representation spaces and $(\boldsymbol{M}_{uv})_{ii} = 0$. Let $\boldsymbol{K}_u$ and $\boldsymbol{K}_v$ denote the kernel matrices associated with the two sites, define the masked alignment statistic

$$\mathcal{A}_{\mathrm{b}}(\boldsymbol{K}, \boldsymbol{L}; \boldsymbol{M}) = \mathcal{H}_{\mathrm{b}}\left(\boldsymbol{K} \odot \boldsymbol{M},\ \boldsymbol{L} \odot \boldsymbol{M}\right), \tag{29}$$

where $\odot$ denotes the Hadamard product. The same pair-specific mask $\boldsymbol{M}_{uv}$ is used in the cross term and in both self-normalization terms. The resulting similarity is then given by

$$s_{\mathrm{CKNNA}}(u,v) = \frac{\mathcal{A}_{\mathrm{b}}(\boldsymbol{K}_u, \boldsymbol{K}_v; \boldsymbol{M}_{uv})}{\sqrt{\mathcal{A}_{\mathrm{b}}(\boldsymbol{K}_u, \boldsymbol{K}_u; \boldsymbol{M}_{uv})\, \mathcal{A}_{\mathrm{b}}(\boldsymbol{K}_v, \boldsymbol{K}_v; \boldsymbol{M}_{uv})}}. \tag{30}$$

The CKNNA dissimilarity is then given by $\delta_{\mathrm{CKNNA}}(u,v) = 1 - s_{\mathrm{CKNNA}}(u,v)$.

Held-out linear-readout disagreement. Held-out linear-readout disagreement compares two representation sites through the predictions of linear classifiers fitted independently to their features (Alain & Bengio, 2016; Harvey et al., 2024).

Let $\mathcal{I}_{\mathrm{fit}}$, $\mathcal{I}_{\mathrm{tune}}$, and $\mathcal{I}_{\mathrm{eval}}$ denote disjoint sets of reference-example indices used for readout fitting, hyperparameter selection, and relation evaluation, respectively. Assume a classification task with C classes having the same semantics and class ordering across all models. At site u, the linear readout has weight matrix $\boldsymbol{W}_u \in \mathbb{R}^{C \times p_u}$ and intercept $\boldsymbol{b}_u \in \mathbb{R}^C$. Let $\widetilde{r}_u(\boldsymbol{x}) \in \mathbb{R}^{p_u}$ denote the representation. The predicted class-probability vector is

$$\boldsymbol{p}_u(\boldsymbol{x}) = \mathrm{softmax}\left(\boldsymbol{W}_u \widetilde{r}_u(\boldsymbol{x}) + \boldsymbol{b}_u\right) \in [0,1]^C. \tag{31}$$

Let $c_i \in \{1, \dots, C\}$ denote the class label of reference example $\boldsymbol{x}_i$. For regularization parameter $\lambda_{\mathrm{ro}} > 0$, we fit the readout by minimizing

$$\mathcal{L}_{\mathrm{ro}}(\boldsymbol{W}_u, \boldsymbol{b}_u) = -\frac{1}{|\mathcal{I}_{\mathrm{fit}}|} \sum_{i \in \mathcal{I}_{\mathrm{fit}}} \log\left[\mathrm{softmax}\left(\boldsymbol{W}_u \widetilde{r}_u(\boldsymbol{x}_i) + \boldsymbol{b}_u\right)\right]_{c_i} + \frac{\lambda_{\mathrm{ro}}}{2} \|\boldsymbol{W}_u\|_F^2. \tag{32}$$

The cross-entropy term is averaged over the fitting examples.

After fitting, the hard prediction of the readout at site u is given by

$$\widehat{c}_u(\boldsymbol{x}) = \underset{c \in \{1,\dots,C\}}{\arg\max} \left[\boldsymbol{p}_u(\boldsymbol{x})\right]_c. \tag{33}$$

We define the held-out readout disagreement between sites u and v as

$$\delta_{\mathrm{ro}}(u,v) = \frac{1}{|\mathcal{I}_{\mathrm{eval}}|} \sum_{i \in \mathcal{I}_{\mathrm{eval}}} \mathbb{I}\left[\widehat{c}_u(\boldsymbol{x}_i) \neq \widehat{c}_v(\boldsymbol{x}_i)\right]. \tag{34}$$

Thus, $\delta_{\mathrm{ro}}(u, v)$ is the fraction of held-out examples on which the two independently fitted readouts predict different classes.

For fixed fitted readouts, let $\boldsymbol{F}_u \in \{0,1\}^{|\mathcal{I}_{\mathrm{eval}}| \times C}$ denote the matrix of one-hot hard predictions at site u, with row i corresponding to the prediction on the ith held-out evaluation example. Then

$$\delta_{\mathrm{ro}}(u, v) = \frac{\|\boldsymbol{F}_u - \boldsymbol{F}_v\|_F^2}{2|\mathcal{I}_{\mathrm{eval}}|}. \tag{35}$$

Hence, for fixed readouts and a fixed evaluation set, hard-readout disagreement is a squared Euclidean distance.

C Theoretical properties

C.1 Symmetries of the site graphs and predictors

We show which changes to a model leave the scores predicted by PARIS unchanged. The proof uses the shared graph updates (Lim et al., 2024a) and our paired attention encoding (Theus et al., 2025). We also show that swapping the two compared sites does not change the predicted score.

Let $G_u = (V_u, E_u, X_u, A_u, T_u)$ denote the graph for site u, where $X_u = (\boldsymbol{x}_i)_{i \in V_u}$ contains the node attributes, $A_u = (\boldsymbol{a}_e)_{e \in E_u}$ contains the edge attributes, and $T_u \subseteq V_u$ marks the nodes associated with the target site. The graph may contain parallel edges. A graph relabeling changes only the node and edge labels. It moves each attribute, endpoint, and target marker together with the corresponding node or edge: $s'(\pi_E(e)) = \pi_V(s(e)), t'(\pi_E(e)) = \pi_V(t(e)), \boldsymbol{x}'_{\pi_V(i)} = \boldsymbol{x}_i, \boldsymbol{a}'_{\pi_E(e)} = \boldsymbol{a}_e, t'_u = \pi_V(t_u)$, where $s(e)$ and $t(e)$ denote the source and target of edge e, and π_V and π_E are bijections. The relabeling does not change node or edge types, operand roles, or spatial attributes. We write $G \cong G'$ when two graphs differ only by such labels.

We call a hidden-coordinate permutation *compatible* if it reorders every affected weight axis, bias, and connection in the same way. The permutation is required to also respect residual branches, normalization groups, shared parameters, and the rule used to extract the target representation. Coordinates with a fixed external meaning remain fixed. Permuting complete attention heads moves each head's QK and VO operators with its dimensions, scale, and mask. The graph does not encode arbitrary coordinate or head indices.

For attention head a, we use row-vector notation and denote

$$\boldsymbol{W}_{Q,a}, \boldsymbol{W}_{K,a} \in \mathbb{R}^{D \times d_{q,a}}, \qquad \boldsymbol{W}_{V,a} \in \mathbb{R}^{D \times d_{v,a}}, \qquad \boldsymbol{W}_{O,a} \in \mathbb{R}^{d_{v,a} \times D}. \tag{36}$$

The biases $\boldsymbol{b}_{Q,a}$, $\boldsymbol{b}_{K,a}$, and $\boldsymbol{b}_{V,a}$ are row vectors, and $\boldsymbol{b}_O \in \mathbb{R}^{1 \times D}$. The graph encodes

$$\begin{aligned} \boldsymbol{C}_a^{\mathrm{QK}} &= \boldsymbol{W}_{Q,a} \boldsymbol{W}_{K,a}^\top, & \boldsymbol{C}_a^{\mathrm{VO}} &= \boldsymbol{W}_{V,a} \boldsymbol{W}_{O,a}, \\ \bar{\boldsymbol{C}}_a^{\mathrm{QK}} &= \alpha_a \begin{bmatrix} \boldsymbol{C}_a^{\mathrm{QK}} \\ \boldsymbol{b}_{Q,a} \boldsymbol{W}_{K,a}^\top \end{bmatrix}, & \boldsymbol{b}_{\mathrm{eff}} &= \boldsymbol{b}_O + \sum_a \boldsymbol{b}_{V,a} \boldsymbol{W}_{O,a}, \end{aligned} \tag{37}$$

where α_a denotes the logit scale. The graph keeps each QK/VO pair in the same head subgraph and records fixed settings such as its scale and mask. We use this encoding for ordinary affine scaled dot-product attention and for target sites at or after the combined projected attention output.

Proposition 3. *Let $\widehat{\boldsymbol{y}}_{\mathsf{A}}(G_u, G_v)$ denote the score vector predicted by $\mathsf{A} \in \{\mathrm{Ind}, \mathrm{Joint}\}$. For fixed metanetwork parameters, the following properties hold.*

(i) ***Graph relabeling.*** *Relabeling either site graph leaves the independent site embedding and the predicted scores unchanged. For any relabelings π_u and π_v,*

$$\mathrm{Enc}_\phi(\pi_u G_u) = \mathrm{Enc}_\phi(G_u), \tag{38}$$

and

$$\widehat{\boldsymbol{y}}_{\mathsf{A}}(\pi_u G_u, \pi_v G_v) = \widehat{\boldsymbol{y}}_{\mathsf{A}}(G_u, G_v). \tag{39}$$

(ii) ***Changes of basis within attention heads.*** *For invertible matrices* $\boldsymbol{A}_a \in \mathbb{R}^{d_{q,a}\times d_{q,a}}$ *and* $\boldsymbol{B}_a \in \mathbb{R}^{d_{v,a}\times d_{v,a}}$*, let*

$$\begin{aligned} \boldsymbol{W}'_{Q,a} &= \boldsymbol{W}_{Q,a}\boldsymbol{A}_a, & \boldsymbol{W}'_{K,a} &= \boldsymbol{W}_{K,a}\boldsymbol{A}_a^{-\top}, \\ \boldsymbol{W}'_{V,a} &= \boldsymbol{W}_{V,a}\boldsymbol{B}_a, & \boldsymbol{W}'_{O,a} &= \boldsymbol{B}_a^{-1}\boldsymbol{W}_{O,a}, \\ \boldsymbol{b}'_{Q,a} &= \boldsymbol{b}_{Q,a}\boldsymbol{A}_a, & \boldsymbol{b}'_{K,a} &= \boldsymbol{b}_{K,a}\boldsymbol{A}_a^{-\top}, \\ \boldsymbol{b}'_{V,a} &= \boldsymbol{b}_{V,a}\boldsymbol{B}_a, & \boldsymbol{b}'_{O} &= \boldsymbol{b}_{O}. \end{aligned} \tag{40}$$

Keep all other parameters, settings, and the target extraction rule fixed. These transformations leave the encoded attention operators unchanged. Applying them to either site then leaves the independent embedding and the scores from both estimators unchanged.

(iii) ***Swapping the sites.*** *Both estimators are symmetric:*

$$\widehat{\boldsymbol{y}}_{\mathsf{A}}(G_u, G_v) = \widehat{\boldsymbol{y}}_{\mathsf{A}}(G_v, G_u). \tag{41}$$

Proof. **Graph relabeling.** Let $G' = \pi G$. Because the shared node encoder receives the same attributes at corresponding nodes, $\boldsymbol{h}'^{(0)}_{\pi_V(i)} = \boldsymbol{h}^{(0)}_i$. Assume that this equality holds after message-passing round r. For an edge e, the message rule gives

$$\begin{aligned} \boldsymbol{m}'^{(r)}_{\pi_E(e)} &= \left(\boldsymbol{W}^{(r)}\boldsymbol{h}'^{(r)}_{\pi_V(s(e))}\right) \odot \psi_{\mathrm{edge},r}(\boldsymbol{a}'_{\pi_E(e)}) \\ &= \left(\boldsymbol{W}^{(r)}\boldsymbol{h}^{(r)}_{s(e)}\right) \odot \psi_{\mathrm{edge},r}(\boldsymbol{a}_e) = \boldsymbol{m}^{(r)}_e. \end{aligned}$$

Every incoming edge of node j has a corresponding edge at $\pi_V(j)$, including parallel and reverse edges. The same holds when a node has no incoming edges. Because the two graphs use the same update MLP, residual step, and nodewise LayerNorm, we obtain $\boldsymbol{h}'^{(r+1)}_{\pi_V(j)} = \boldsymbol{h}^{(r+1)}_j$. Induction gives then this equality after every message-passing round.

Relabeling also leaves the two averages used by PARIS-Ind unchanged:

$$\boldsymbol{p}'_u = \frac{1}{|\pi_V(t_u)|}\sum_{i'\in\pi_V(t_u)} \boldsymbol{h}'^{(R)}_{i'} = \frac{1}{|t_u|}\sum_{i\in t_u}\boldsymbol{h}^{(R)}_i = \boldsymbol{p}_u,$$

$$\boldsymbol{g}'_u = \frac{1}{|V'_u|}\sum_{i'\in V'_u} \boldsymbol{h}'^{(R)}_{i'} = \frac{1}{|V_u|}\sum_{i\in V_u}\boldsymbol{h}^{(R)}_i = \boldsymbol{g}_u.$$

Thus, the readout $\boldsymbol{z}_u = \rho_\phi([\boldsymbol{p}_u \| \boldsymbol{g}_u])$ receives the same input and returns the same site embedding. Each comparison head in PARIS-Ind returns the same score.

For PARIS-Joint, relabeling only reorders the operator and head tokens. It leaves the summary tokens and token roles unchanged. For one cross-attention head, we have

$$\mathrm{CA}_{\boldsymbol{M}}(\boldsymbol{U},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\mathcal{Q}(\boldsymbol{U})\mathcal{K}(\boldsymbol{V})^\top}{\sqrt{d_a}} + \boldsymbol{M}\right)\mathcal{V}(\boldsymbol{V}),$$

where $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ are affine maps applied to each row. These maps commute with row permutations. For permutation matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ and masked logits $\boldsymbol{L}$,

$$\mathrm{softmax}(\boldsymbol{P}_1\boldsymbol{L}\boldsymbol{P}_2^\top) = \boldsymbol{P}_1\,\mathrm{softmax}(\boldsymbol{L})\boldsymbol{P}_2^\top.$$

For masks containing $-\infty$, a permutation only reindexes the entries. Using $\boldsymbol{P}_2^\top\boldsymbol{P}_2 = \boldsymbol{I}$ gives

$$\mathrm{CA}_{\boldsymbol{P}_1\boldsymbol{M}\boldsymbol{P}_2^\top}(\boldsymbol{P}_1\boldsymbol{U},\boldsymbol{P}_2\boldsymbol{V}) = \boldsymbol{P}_1\,\mathrm{CA}_M(\boldsymbol{U},\boldsymbol{V}).$$

Thus, permuting the query and key/value tokens only permutes the output query tokens in the same way. The output projection, residual update, feed-forward network, and LayerNorm preserve this property because they apply the same function to corresponding rows. Masks remove padded keys and pooling ignores padded tokens. The target-summary token keeps its role, and averaging valid tokens is independent of their order. The two contextual readouts then return the same vectors, so the joint estimator also returns the same score.

We now connect graph relabeling to hidden-coordinate permutations. For this example only, use column-vector notation. For two linear maps separated by a coordinatewise nonlinearity, a permutation $\boldsymbol{P}$ of the hidden coordinates gives

$$(\boldsymbol{W}_1, \boldsymbol{b}_1, \boldsymbol{W}_2) \longmapsto (\boldsymbol{P}\boldsymbol{W}_1, \boldsymbol{P}\boldsymbol{b}_1, \boldsymbol{W}_2\boldsymbol{P}^\top).$$

Thus, the corresponding rows of $\boldsymbol{W}_1$, entries of $\boldsymbol{b}_1$, and columns of $\boldsymbol{W}_2$ move together. This operation only relabels the intermediate coordinate nodes. The same consistency holds across residual branches, normalization groups, and the target extraction rule.

The attention encoding behaves in the same way under coordinate permutations. If $\boldsymbol{P}$ and $\boldsymbol{R}$ permute the input and output residual coordinates, then we have

$$\boldsymbol{C}_a^{\mathrm{QK}\prime} = \boldsymbol{P}\boldsymbol{C}_a^{\mathrm{QK}}\boldsymbol{P}^\top, \qquad \boldsymbol{C}_a^{\mathrm{VO}\prime} = \boldsymbol{P}\boldsymbol{C}_a^{\mathrm{VO}}\boldsymbol{R}^\top.$$

The query-bias term becomes $\boldsymbol{b}_{Q,a}\boldsymbol{W}_{K,a}^\top\boldsymbol{P}^\top$, and the effective output bias becomes $\boldsymbol{b}_{\mathrm{eff}}\boldsymbol{R}^\top$. Permuting complete attention heads also moves each paired head subgraph as one unit. All these transformations reduce to graph relabelings, so the result above applies.

Changes of basis within attention heads. We omit the head index. Substituting the transformed parameters from Eq. (40) gives

$$\begin{aligned}
\boldsymbol{W}_Q'(\boldsymbol{W}_K')^\top &= \boldsymbol{W}_Q\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{W}_K^\top = \boldsymbol{C}^{\mathrm{QK}},\\
\boldsymbol{W}_V'\boldsymbol{W}_O' &= \boldsymbol{W}_V\boldsymbol{B}\boldsymbol{B}^{-1}\boldsymbol{W}_O = \boldsymbol{C}^{\mathrm{VO}},\\
\boldsymbol{b}_Q'(\boldsymbol{W}_K')^\top &= \boldsymbol{b}_Q\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{W}_K^\top = \boldsymbol{b}_Q\boldsymbol{W}_K^\top,\\
\boldsymbol{b}_V'\boldsymbol{W}_O' &= \boldsymbol{b}_V\boldsymbol{B}\boldsymbol{B}^{-1}W_O = \boldsymbol{b}_V\boldsymbol{W}_O.
\end{aligned}$$

Hence $\bar{\boldsymbol{C}}_a^{\mathrm{QK}}$, $\boldsymbol{C}_a^{\mathrm{VO}}$, and $\boldsymbol{b}_{\mathrm{eff}}$ remain unchanged. The head pairing, dimensions, scales, masks, and structural edges also remain fixed. Because the graph does not contain the raw projection factors or any basis-dependent statistics, the encoded attention subgraph does not change.

These encoded quantities also determine the combined attention output. For $\boldsymbol{X} \in \mathbb{R}^{T\times D}$, write

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q + \mathbf{1}\boldsymbol{b}_Q, \qquad \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K + \mathbf{1}\boldsymbol{b}_K, \qquad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V + \mathbf{1}\boldsymbol{b}_V.$$

Expanding the logits gives

$$\boldsymbol{Q}\boldsymbol{K}^\top = \boldsymbol{X}\boldsymbol{C}^{\mathrm{QK}}\boldsymbol{X}^\top + \mathbf{1}(\boldsymbol{b}_Q\boldsymbol{W}_K^\top)\boldsymbol{X}^\top + \left(\boldsymbol{X}\boldsymbol{W}_Q\boldsymbol{b}_K^\top + \mathbf{1}\boldsymbol{b}_Q\boldsymbol{b}_K^\top\right)\mathbf{1}^\top.$$

The final term is constant across each query row, so row-wise softmax removes it, including in the presence of a fixed mask. Each row of the resulting attention matrix $P_a(\boldsymbol{X})$ sums to one because every query has at least one allowed key. Thus, $P_a(\boldsymbol{X})\mathbf{1} = \mathbf{1}$. The supported changes of basis leave this tensor unchanged for every input $\boldsymbol{X}$.

Swapping the sites. For PARIS-Ind, each comparison head computes

$$\widehat{y}_{\mathrm{Ind}}^{(h)}(G_u, G_v) = q_h\left(\|\boldsymbol{W}_h(\boldsymbol{z}_u - \boldsymbol{z}_v)\|_2\right).$$

Swapping u and v negates the projected difference but leaves its Euclidean norm unchanged. The calibration then returns the same score.

For PARIS-Joint, let $F_\ell(\boldsymbol{U}, \boldsymbol{V})$ denote the full update of the first stream in cross-attention block ℓ. Both directions use the same function and read the states from the start of the block:

$$\begin{aligned}
\boldsymbol{T}_u^{(\ell+1)} &= F_\ell(\boldsymbol{T}_u^{(\ell)}, \boldsymbol{T}_v^{(\ell)}),\\
\boldsymbol{T}_v^{(\ell+1)} &= F_\ell(\boldsymbol{T}_v^{(\ell)}, \boldsymbol{T}_u^{(\ell)}).
\end{aligned}$$

Swapping the input graphs swaps the initial streams because both sides use the same encoder and tokenization. The masks move with their streams. By induction that every cross-attention block swaps its two outputs. The shared readout swaps $\boldsymbol{c}_{u|v}$ and $\boldsymbol{c}_{v|u}$. The final comparison head uses

$$[\boldsymbol{c}_{u|v} + \boldsymbol{c}_{v|u} \parallel |\boldsymbol{c}_{u|v} - \boldsymbol{c}_{v|u}| \parallel \boldsymbol{c}_{u|v} \odot \boldsymbol{c}_{v|u}],$$

which is unchanged when the two contextual vectors are swapped. The final MLP and sigmoid return the same score. This argument requires the two cross-attention directions to share their parameters, but it does not require the query and key projections within a direction to be equal.

□

C.2 Geometry of independent and joint scores

For a fixed common evaluation sample, the relations considered below admit sitewise Euclidean or squared-Euclidean representations. These constructions motivate embeddings that are computed

without reference to the comparison partner. Our graph metanetwork aims to learn embeddings with corresponding pairwise geometry from the site-specific computation graphs.

Proposition 4. *Given a common sample of n inputs and let $\boldsymbol{C}_n = \boldsymbol{I}_n - \mathbf{1}\mathbf{1}^\top/n$. Let $\boldsymbol{K}_u = \boldsymbol{C}_n \boldsymbol{H}_u \boldsymbol{H}_u^\top \boldsymbol{C}_n$ and assume $\boldsymbol{K}_u \neq 0$ for every site under consideration. For the empirical centered linear CKA defined by normalized Frobenius inner products of centered Gram matrices, set $\widetilde{\boldsymbol{K}}_u = \frac{\boldsymbol{K}_u}{\|\boldsymbol{K}_u\|_F}$ and $\delta_{\mathrm{C}}(u,v) = \sqrt{1 - \langle \widetilde{\boldsymbol{K}}_u, \widetilde{\boldsymbol{K}}_v \rangle_F}$. Then, we have $\delta_{\mathrm{C}}(u,v) = \|\Phi_{\mathrm{C}}(u) - \Phi_{\mathrm{C}}(v)\|_2$ and $\Phi_{\mathrm{C}}(u) = 2^{-1/2}\,\mathrm{vec}(\widetilde{\boldsymbol{K}}_u)$. In addition, let $\overline{\boldsymbol{H}}_u$ denote the activation matrix obtained using the fixed sitewise centering and scaling convention used to define the GULP targets. Define $\boldsymbol{A}_u = \frac{\overline{\boldsymbol{H}}_u}{\sqrt{n}}$, $\boldsymbol{S}_u = \boldsymbol{A}_u^\top \boldsymbol{A}_u$, $\boldsymbol{S}_{uv} = \boldsymbol{A}_u^\top \boldsymbol{A}_v$, and $\boldsymbol{R}_u = (\boldsymbol{S}_u + \lambda \boldsymbol{I})^{-1}$ for $\lambda > 0$. For the covariance form of GULP, we have $\delta_{\mathrm{G},\lambda}(u,v)^2 = \mathrm{tr}(\boldsymbol{R}_u \boldsymbol{S}_u \boldsymbol{R}_u \boldsymbol{S}_u) + \mathrm{tr}(\boldsymbol{R}_v \boldsymbol{S}_v \boldsymbol{R}_v \boldsymbol{S}_v) - 2\,\mathrm{tr}(\boldsymbol{R}_u \boldsymbol{S}_{uv} \boldsymbol{R}_v \boldsymbol{S}_{uv}^\top)$, and $\delta_{\mathrm{G},\lambda}(u,v) = \|\Phi_{\mathrm{G},\lambda}(u) - \Phi_{\mathrm{G},\lambda}(v)\|_2$, where $\Phi_{\mathrm{G},\lambda}(u) = \mathrm{vec}(\boldsymbol{A}_u \boldsymbol{R}_u \boldsymbol{A}_u^\top)$.*

Proof. We treat the CKA and GULP cases separately.

First, for CKA, observe that $\boldsymbol{K}_u = \boldsymbol{C}_n \boldsymbol{H}_u \boldsymbol{H}_u^\top \boldsymbol{C}_n = (\boldsymbol{C}_n \boldsymbol{H}_u)(\boldsymbol{C}_n \boldsymbol{H}_u)^\top$, because $\boldsymbol{C}_n$ is symmetric. Thus, $\boldsymbol{K}_u$ is symmetric positive semidefinite. Since $\boldsymbol{K}_u \neq 0$, its Frobenius normalization is well defined and satisfies $\|\widetilde{\boldsymbol{K}}_u\|_F = 1$. The same statements hold for the site v. Moreover, since both normalized Gram matrices are positive semidefinite, we have $\left\langle \widetilde{\boldsymbol{K}}_u, \widetilde{\boldsymbol{K}}_v \right\rangle_F = \mathrm{tr}\left(\widetilde{\boldsymbol{K}}_u \widetilde{\boldsymbol{K}}_v\right) \geq 0$. Note that $\mathrm{tr}\left(\widetilde{\boldsymbol{K}}_u \widetilde{\boldsymbol{K}}_v\right) = \mathrm{tr}\left(\widetilde{\boldsymbol{K}}_u^{1/2} \widetilde{\boldsymbol{K}}_v \widetilde{\boldsymbol{K}}_u^{1/2}\right) \geq 0$, because the matrix inside the last trace is positive semidefinite. In addition, by the Frobenius Cauchy-Schwarz inequality, we have $\left\langle \widetilde{\boldsymbol{K}}_u, \widetilde{\boldsymbol{K}}_v \right\rangle_F \leq \|\widetilde{\boldsymbol{K}}_u\|_F \|\widetilde{\boldsymbol{K}}_v\|_F = 1$. Thus, the normalized Frobenius inner product lies in $[0,1]$, so δ_{C} is well defined. We now expand the squared Frobenius distance between the two normalized Gram matrices:

$$\begin{aligned}
\frac{1}{2}\|\widetilde{\boldsymbol{K}}_u - \widetilde{\boldsymbol{K}}_v\|_F^2 &= \frac{1}{2}\left(\|\widetilde{\boldsymbol{K}}_u\|_F^2 + \|\widetilde{\boldsymbol{K}}_v\|_F^2 - 2\left\langle \widetilde{\boldsymbol{K}}_u, \widetilde{\boldsymbol{K}}_v \right\rangle_F\right) \\
&= 1 - \left\langle \widetilde{\boldsymbol{K}}_u, \widetilde{\boldsymbol{K}}_v \right\rangle_F \\
&= \delta_{\mathrm{C}}(u,v)^2.
\end{aligned}$$

Therefore, we obtain

$$\delta_{\mathrm{C}}(u,v) = \frac{1}{\sqrt{2}}\|\widetilde{\boldsymbol{K}}_u - \widetilde{\boldsymbol{K}}_v\|_F.$$

Finally, vectorization is an isometry between the Frobenius norm on matrices and the Euclidean norm on their vectorizations, i.e., $\|\,\mathrm{vec}(\boldsymbol{X})\|_2 = \|\boldsymbol{X}\|_F$. Thus, we have

$$\delta_{\mathrm{C}}(u,v) = \left\| \frac{1}{\sqrt{2}}\,\mathrm{vec}(\widetilde{\boldsymbol{K}}_u) - \frac{1}{\sqrt{2}}\,\mathrm{vec}(\widetilde{\boldsymbol{K}}_v) \right\|_2 = \|\Phi_{\mathrm{C}}(u) - \Phi_{\mathrm{C}}(v)\|_2.$$

We next show that the covariance expression for GULP is the squared Frobenius distance between two $n \times n$ matrices. Since $\boldsymbol{S}_u = \boldsymbol{A}_u^\top \boldsymbol{A}_u$ is symmetric positive semidefinite and $\lambda > 0$, we have, for every nonzero $\boldsymbol{x} \in \mathbb{R}^{p_u}$ that $\boldsymbol{x}^\top(\boldsymbol{S}_u + \lambda \boldsymbol{I}_{p_u})\boldsymbol{x} = \|\boldsymbol{A}_u \boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_2^2 > 0$. Thus, $\boldsymbol{S}_u + \lambda \boldsymbol{I}_{p_u}$ is positive definite and invertible. In addition, $\boldsymbol{R}_u$ is well defined and symmetric. Next, define the sample-space matrix $\boldsymbol{T}_u := \boldsymbol{A}_u \boldsymbol{R}_u \boldsymbol{A}_u^\top \in \mathbb{R}^{n \times n}$. Since $\boldsymbol{R}_u$ is symmetric, we have $\boldsymbol{T}_u^\top = \boldsymbol{A}_u \boldsymbol{R}_u^\top \boldsymbol{A}_u^\top = \boldsymbol{T}_u$, and thus $\boldsymbol{T}_u$ is symmetric. Note that since $\boldsymbol{T}_u$ is always an $n \times n$ matrix, irrespective of the representation dimension p_u. Thus $\boldsymbol{T}_u$ and $\boldsymbol{T}_v$ belong to the same sample space even when $p_u \neq p_v$.

We then compute the squared Frobenius norm of $\boldsymbol{T}_u$. Since $\boldsymbol{T}_u$ is symmetric, we have

$$\|\boldsymbol{T}_u\|_F^2 = \mathrm{tr}(\boldsymbol{T}_u^\top \boldsymbol{T}_u) = \mathrm{tr}(\boldsymbol{T}_u^2) = \mathrm{tr}\left(\boldsymbol{A}_u \boldsymbol{R}_u \boldsymbol{A}_u^\top \boldsymbol{A}_u \boldsymbol{R}_u \boldsymbol{A}_u^\top\right).$$

Using cyclicity of the trace and $\boldsymbol{A}_u^\top \boldsymbol{A}_u = \boldsymbol{S}_u$, this becomes

$$\|\boldsymbol{T}_u\|_F^2 = \mathrm{tr}\left(\boldsymbol{R}_u \boldsymbol{A}_u^\top \boldsymbol{A}_u \boldsymbol{R}_u \boldsymbol{A}_u^\top \boldsymbol{A}_u\right) = \mathrm{tr}(\boldsymbol{R}_u \boldsymbol{S}_u \boldsymbol{R}_u \boldsymbol{S}_u).$$

Similarly, we have $\|\boldsymbol{T}_v\|_F^2 = \operatorname{tr}(\boldsymbol{R}_v\boldsymbol{S}_v\boldsymbol{R}_v\boldsymbol{S}_v)$. For the cross term, by using symmetry and cyclicity of the trace, we have

$$\begin{aligned}\langle \boldsymbol{T}_u, \boldsymbol{T}_v\rangle_F &= \operatorname{tr}(\boldsymbol{T}_u^\top \boldsymbol{T}_v) = \operatorname{tr}(\boldsymbol{T}_u\boldsymbol{T}_v)\\ &= \operatorname{tr}\left(\boldsymbol{A}_u\boldsymbol{R}_u\boldsymbol{A}_u^\top\boldsymbol{A}_v\boldsymbol{R}_v\boldsymbol{A}_v^\top\right)\\ &= \operatorname{tr}\left(\boldsymbol{R}_u\boldsymbol{A}_u^\top\boldsymbol{A}_v\boldsymbol{R}_v\boldsymbol{A}_v^\top\boldsymbol{A}_u\right).\end{aligned}$$

Since $\boldsymbol{A}_u^\top\boldsymbol{A}_v = \boldsymbol{S}_{uv}$ and $\boldsymbol{A}_v^\top\boldsymbol{A}_u = \boldsymbol{S}_{uv}^\top$, we have $\langle \boldsymbol{T}_u, \boldsymbol{T}_v\rangle_F = \operatorname{tr}\left(\boldsymbol{R}_u\boldsymbol{S}_{uv}\boldsymbol{R}_v\boldsymbol{S}_{uv}^\top\right)$. Thus, we can expand the squared Frobenius distance as

$$\begin{aligned}\|\boldsymbol{T}_u - \boldsymbol{T}_v\|_F^2 &= \|\boldsymbol{T}_u\|_F^2 + \|\boldsymbol{T}_v\|_F^2 - 2\langle \boldsymbol{T}_u, \boldsymbol{T}_v\rangle_F\\ &= \operatorname{tr}(\boldsymbol{R}_u\boldsymbol{S}_u\boldsymbol{R}_u\boldsymbol{S}_u) + \operatorname{tr}(\boldsymbol{R}_v\boldsymbol{S}_v\boldsymbol{R}_v\boldsymbol{S}_v) - 2\operatorname{tr}\left(\boldsymbol{R}_u\boldsymbol{S}_{uv}\boldsymbol{R}_v\boldsymbol{S}_{uv}^\top\right)\\ &= \delta_{\mathrm{G},\lambda}(u,v)^2.\end{aligned}$$

Both sides are nonnegative, by taking square roots gives $\delta_{\mathrm{G},\lambda}(u,v) = \|\boldsymbol{T}_u - \boldsymbol{T}_v\|_F$. Applying the isometry of vectorization, we have

$$\delta_{\mathrm{G},\lambda}(u,v) = \|\operatorname{vec}(\boldsymbol{T}_u) - \operatorname{vec}(\boldsymbol{T}_v)\|_2 = \|\Phi_{\mathrm{G},\lambda}(u) - \Phi_{\mathrm{G},\lambda}(v)\|_2,$$

where $\Phi_{\mathrm{G},\lambda}(u) = \operatorname{vec}(\boldsymbol{A}_u\boldsymbol{R}_u\boldsymbol{A}_u^\top)$. □

Proposition 5. *Let $N \geq 2$ and $k \geq 1$. Let $\boldsymbol{Y} \in \mathbb{R}^{N\times N}$ be symmetric, entrywise nonnegative, and zero on the diagonal. Let $q : [0,\infty) \to [0,\infty)$ be strictly increasing with $q(0) = 0$, and assume that $\boldsymbol{Y}_{ij} \in q([0,\infty))$ for all i, j. Since q is strictly increasing, its inverse is well defined on its range. Define $\boldsymbol{D}_{ij} = q^{-1}(\boldsymbol{Y}_{ij})$, let $\boldsymbol{J} = \boldsymbol{I}_N - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$, and define the double-centered matrix $\boldsymbol{B}_{\mathrm{D}} = -\frac{1}{2}\boldsymbol{J}\boldsymbol{D}^{\circ 2}\boldsymbol{J}$, where $\boldsymbol{D}^{\circ 2}$ denotes the entrywise square of $\boldsymbol{D}$. Then, there exist vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N \in \mathbb{R}^k$ such that $\boldsymbol{Y}_{ij} = q(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2)$ for all i, j if and only if $\boldsymbol{B}_{\mathrm{D}} \succeq 0$ and $\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) \leq k$. That is, the calibrated target matrix $\boldsymbol{Y}$ is realizable by Euclidean site vectors in dimension at most k precisely when the inverse-calibrated matrix $\boldsymbol{D} = q^{-1}(\boldsymbol{Y})$ is a Euclidean distance matrix of embedding dimension at most k.*

Proof. Because q is strictly increasing and $q(0) = 0$, the inverse q^{-1} is well defined on $q([0,\infty))$. Note that $\boldsymbol{D}$ is symmetric and entrywise nonnegative, thus we have $\boldsymbol{D}_{ii} = q^{-1}(\boldsymbol{Y}_{ii}) = q^{-1}(0) = 0$. Suppose there exist $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N \in \mathbb{R}^k$ such that $\boldsymbol{Y}_{ij} = q(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2)$ for every i, j. Applying q^{-1} gives $\boldsymbol{D}_{ij} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2$. Let $\bar{\boldsymbol{v}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{v}_i$, $\widetilde{\boldsymbol{v}}_i = \boldsymbol{v}_i - \bar{\boldsymbol{v}}$, and let $\boldsymbol{V} = \begin{pmatrix}\widetilde{\boldsymbol{v}}_1^\top\\ \vdots\\ \widetilde{\boldsymbol{v}}_N^\top\end{pmatrix} \in \mathbb{R}^{N\times k}$. Since translation does not change pairwise distances, we have $\boldsymbol{D}_{ij} = \|\widetilde{\boldsymbol{v}}_i - \widetilde{\boldsymbol{v}}_j\|_2$, in addition, $\mathbf{1}^\top\boldsymbol{V} = 0$. Now set $\boldsymbol{G} = \boldsymbol{V}\boldsymbol{V}^\top$, we then have $\boldsymbol{D}_{ij}^2 = \boldsymbol{G}_{ii} + \boldsymbol{G}_{jj} - 2\boldsymbol{G}_{ij}$. Denote $\boldsymbol{g} = \operatorname{diag}(\boldsymbol{G})$ and denote in matrix form as $\boldsymbol{D}^{\circ 2} = \boldsymbol{g}\mathbf{1}^\top + \mathbf{1}\boldsymbol{g}^\top - 2\boldsymbol{G}$. Since $\boldsymbol{J}\mathbf{1} = 0$, double centering (Schoenberg, 1935) eliminates the first two terms, i.e., $\boldsymbol{J}\boldsymbol{D}^{\circ 2}\boldsymbol{J} = -2\boldsymbol{J}\boldsymbol{G}\boldsymbol{J}$. Since $\boldsymbol{V}$ is centered, we have $\boldsymbol{J}\boldsymbol{V} = \boldsymbol{V}$, and $\boldsymbol{J}\boldsymbol{G}\boldsymbol{J} = \boldsymbol{J}\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{J} = \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{G}$. Thus, we can obtain $\boldsymbol{B}_{\mathrm{D}} = -\frac{1}{2}\boldsymbol{J}\boldsymbol{D}^{\circ 2}\boldsymbol{J} = \boldsymbol{V}\boldsymbol{V}^\top$, $\boldsymbol{B}_{\mathrm{D}} \succeq 0$ and $\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) = \operatorname{rank}(\boldsymbol{V}\boldsymbol{V}^\top) \leq \operatorname{rank}(\boldsymbol{V}) \leq k$.

For another direction, now suppose $\boldsymbol{B}_{\mathrm{D}} \succeq 0$ and $\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) \leq k$. By positive semidefiniteness and the rank bound, there exists $\boldsymbol{V} \in \mathbb{R}^{N\times k}$ such that $\boldsymbol{B}_{\mathrm{D}} = \boldsymbol{V}\boldsymbol{V}^\top$, where zero columns may be added if $\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) < k$. Since $\boldsymbol{B}_{\mathrm{D}} = -\frac{1}{2}\boldsymbol{J}\boldsymbol{D}^{\circ 2}\boldsymbol{J}$ and $\boldsymbol{J}\mathbf{1} = 0$, we have $\boldsymbol{B}_{\mathrm{D}}\mathbf{1} = 0$. It also follows that $0 = \mathbf{1}^\top\boldsymbol{B}_{\mathrm{D}}\mathbf{1} = \mathbf{1}^\top\boldsymbol{V}\boldsymbol{V}^\top\mathbf{1} = \|\boldsymbol{V}^\top\mathbf{1}\|_2^2$. Therefore, we have $\boldsymbol{V}^\top\mathbf{1} = 0$, and the rows of $\boldsymbol{V}$ are centered. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ denote the rows of $\boldsymbol{V}$, and let $\boldsymbol{S}'$ be their squared-distance matrix in the form of $\boldsymbol{S}'_{ij} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$. Let $\boldsymbol{S} = \boldsymbol{D}^{\circ 2}$. Since $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{B}_{\mathrm{D}}$ and the rows of $\boldsymbol{V}$ are centered, by double-centering, we have $\boldsymbol{J}\boldsymbol{S}'\boldsymbol{J} = -2\boldsymbol{B}_{\mathrm{D}}$. By the definition of $\boldsymbol{B}_{\mathrm{D}}$, we have $\boldsymbol{J}\boldsymbol{S}\boldsymbol{J} = -2\boldsymbol{B}_{\mathrm{D}}$., and thus $\boldsymbol{J}\boldsymbol{S}'\boldsymbol{J} = \boldsymbol{J}\boldsymbol{S}\boldsymbol{J}$. If $\boldsymbol{A} := \boldsymbol{S} - \boldsymbol{S}'$, then $\boldsymbol{J}\boldsymbol{A}\boldsymbol{J} = 0$. Moreover, A is symmetric, $\boldsymbol{S}_{ii} = \boldsymbol{D}_{ii}^2 = 0$, and $\boldsymbol{S}'_{ii} = \|\boldsymbol{v}_i - \boldsymbol{v}_i\|_2^2 = 0$. Next, let $\boldsymbol{P} = \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ and $\boldsymbol{J} = \boldsymbol{I}_N - \boldsymbol{P}$. Since $\boldsymbol{J}\boldsymbol{A}\boldsymbol{J} = 0$, we have

$$\begin{aligned}\boldsymbol{A} &= (\boldsymbol{P} + \boldsymbol{J})\boldsymbol{A}(\boldsymbol{P} + \boldsymbol{J})\\ &= \boldsymbol{P}\boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}\boldsymbol{J} + \boldsymbol{J}\boldsymbol{A}\boldsymbol{P}\\ &= \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{A}\boldsymbol{P}.\end{aligned}$$

Next we define $\boldsymbol{r} = \frac{1}{N}\boldsymbol{A}\mathbf{1}$ and $\alpha = \frac{1}{N^2}\mathbf{1}^\top \boldsymbol{A}\mathbf{1}$. Using the symmetry of $\boldsymbol{A}$, we have $\boldsymbol{AP} = \boldsymbol{r}\mathbf{1}^\top$, $\boldsymbol{PA} = \mathbf{1}\boldsymbol{r}^\top$, and $\boldsymbol{PAP} = \alpha\mathbf{1}\mathbf{1}^\top$. Thus, we obtain $\boldsymbol{A} = \mathbf{1}\boldsymbol{r}^\top + \boldsymbol{r}\mathbf{1}^\top - \alpha\mathbf{1}\mathbf{1}^\top$. By setting $\boldsymbol{a} = \boldsymbol{r} - \frac{\alpha}{2}\mathbf{1}$, we obtain $\boldsymbol{A} = \mathbf{1}\boldsymbol{a}^\top + \boldsymbol{a}\mathbf{1}^\top$. Its diagonal entries satisfy $\boldsymbol{A}_{ii} = 2\boldsymbol{a}_i$. But $\boldsymbol{A}_{ii} = 0$ for every i, thus $\boldsymbol{a}_i = 0$ for every i, and $\boldsymbol{A} = 0$. Thus, we have $\boldsymbol{S} = \boldsymbol{S}'$, that is, $\boldsymbol{D}_{ij}^2 = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$ for all i, j. Both sides are nonnegative, and $\boldsymbol{D}_{ij} \geq 0$, we obtain $\boldsymbol{D}_{ij} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2$. Finally, applying q to both sides, we get $\boldsymbol{Y}_{ij} = q(\boldsymbol{D}_{ij}) = q\left(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2\right)$ for all i, j. □

Corollary 6. *There exist three valid centered activation matrices $\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3$ whose pairwise chordal-CKA distances cannot be represented in the form $1 - \exp\left(-\varepsilon\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2\right)$, for $i, j \in \{1, 2, 3\}$, for any $\varepsilon > 0$, any finite embedding dimension $k \geq 1$, and any vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{R}^k$. Specifically, let $\boldsymbol{Y}_{ij} = \delta_{\mathrm{C}}(\boldsymbol{H}_i, \boldsymbol{H}_j)$ denote their chordal-CKA distances, and define*

$$E_\star := \inf_{\substack{\varepsilon>0,\ k\geq 1 \\ \boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3\in\mathbb{R}^k}} \max_{1\leq i<j\leq 3} \left|1 - \exp\left(-\varepsilon\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2\right) - \boldsymbol{Y}_{ij}\right|. \tag{42}$$

Then $E_\star = E_0 := 1 - \frac{1}{2}\sqrt{7 - 2\sqrt{3}} > 0$. The three chordal-CKA distances admit no exact exponentially calibrated Euclidean realization, regardless of the embedding dimension.

Proof. We first construct the three centered activation matrices and compute their chordal-CKA distances. Choose orthonormal vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3$ that are orthogonal to $\mathbf{1}$. For example, $\boldsymbol{a} = \frac{1}{\sqrt{2}}(1, -1, 0)^\top$ and $\boldsymbol{b} = \frac{1}{\sqrt{6}}(1, 1, -2)^\top$. For $\theta \in \left\{0, \frac{\pi}{6}, \frac{\pi}{3}\right\}$, define the single-column activation matrix as $\boldsymbol{H}_\theta = \cos\theta\, \boldsymbol{a} + \sin\theta\, \boldsymbol{b} \in \mathbb{R}^{3\times 1}$. Since $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthogonal to $\mathbf{1}$, we have $\mathbf{1}^\top \boldsymbol{H}_\theta = 0$, and thus $\boldsymbol{H}_\theta$ is centered. Moreover, since $\boldsymbol{a}$ and $\boldsymbol{b}$ are orthonormal, we have $\|\boldsymbol{H}_\theta\|_2 = 1$, and the centered Gram matrix is $\boldsymbol{K}_\theta = \boldsymbol{H}_\theta \boldsymbol{H}_\theta^\top$, and $\|\boldsymbol{K}_\theta\|_F = \|\boldsymbol{H}_\theta\|_2^2 = 1$. For two angles θ, θ', we consider

$$\langle \boldsymbol{K}_\theta, \boldsymbol{K}_{\theta'} \rangle_F = \operatorname{tr}\left(\boldsymbol{H}_\theta \boldsymbol{H}_\theta^\top \boldsymbol{H}_{\theta'} \boldsymbol{H}_{\theta'}^\top\right) = \left(\boldsymbol{H}_\theta^\top \boldsymbol{H}_{\theta'}\right)^2 = \cos^2(\theta - \theta').$$

Then, we have $\mathrm{CKA}(\boldsymbol{H}_\theta, \boldsymbol{H}_{\theta'}) = \cos^2(\theta - \theta')$, and the corresponding chordal-CKA distance is $\delta_{\mathrm{C}}(\boldsymbol{H}_\theta, \boldsymbol{H}_{\theta'}) = \sqrt{1 - \cos^2(\theta - \theta')} = |\sin(\theta - \theta')|$. Now we index the three matrices by $\boldsymbol{H}_1 = \boldsymbol{H}_0$, $\boldsymbol{H}_2 = \boldsymbol{H}_{\pi/6}$ and $\boldsymbol{H}_3 = \boldsymbol{H}_{\pi/3}$. Their pairwise target distances are respectively $\boldsymbol{Y}_{12} = \boldsymbol{Y}_{23} = \frac{1}{2}$, and $\boldsymbol{Y}_{13} = c := \frac{\sqrt{3}}{2}$. We next derive a necessary constraint on any exponentially calibrated Euclidean realization. Fix arbitrary $\varepsilon > 0$, $k \geq 1$, and $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{R}^k$, and let $\widehat{\boldsymbol{Y}}_{ij} = 1 - \exp\left(-\varepsilon\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2\right)$. Let $d_{ij} = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2$. By the Euclidean triangle inequality, we have $d_{13} \leq d_{12} + d_{23}$. Since $q_\varepsilon(t) = 1 - e^{-\varepsilon t}$ is strictly increasing, we have

$$\widehat{\boldsymbol{Y}}_{13} \leq q_\varepsilon(d_{12} + d_{23}).$$

For $s, t \geq 0$, we have

$$\begin{aligned} q_\varepsilon(s + t) &= 1 - e^{-\varepsilon(s+t)} \\ &= 1 - (1 - q_\varepsilon(s))(1 - q_\varepsilon(t)) \\ &= q_\varepsilon(s) + q_\varepsilon(t) - q_\varepsilon(s) q_\varepsilon(t). \end{aligned}$$

Thus, every exponentially calibrated Euclidean realization need to satisfy $\widehat{\boldsymbol{Y}}_{13} \leq \widehat{\boldsymbol{Y}}_{12} + \widehat{\boldsymbol{Y}}_{23} - \widehat{\boldsymbol{Y}}_{12}\widehat{\boldsymbol{Y}}_{23}$. At the target values in the CKA counterexample targets, the right-hand side would equal $\frac{1}{2} + \frac{1}{2} - \frac{1}{4} = \frac{3}{4}$, whereas $\boldsymbol{Y}_{13} = \frac{\sqrt{3}}{2} > \frac{3}{4}$. Therefore, exact realization is impossible. For an arbitrary candidate realization, set $E = \max_{1\leq i<j\leq 3} |\widehat{\boldsymbol{Y}}_{ij} - \boldsymbol{Y}_{ij}|$. We show that necessarily $E \geq E_0$. If $E \geq 1/2$, then the desired conclusion follows immediately because $E_0 < 1/2$. Suppose that $E < \frac{1}{2}$. The definition of E implies that $\widehat{\boldsymbol{Y}}_{12} \leq \frac{1}{2} + E$, $\widehat{\boldsymbol{Y}}_{23} \leq \frac{1}{2} + E$, and $\widehat{\boldsymbol{Y}}_{13} \geq c - E$. Since $E < 1/2$, both $\widehat{\boldsymbol{Y}}_{12}$ and $\widehat{\boldsymbol{Y}}_{23}$ lie in $[0, 1)$. The function $f(x, y) = x + y - xy$ is nondecreasing in each argument on $[0, 1]^2$, since $\frac{\partial f}{\partial x} = 1 - y \geq 0$ and $\frac{\partial f}{\partial y} = 1 - x \geq 0$. Therefore, combining all these, we have

$$\begin{aligned} c - E &\leq \widehat{\boldsymbol{Y}}_{13} \\ &\leq \widehat{\boldsymbol{Y}}_{12} + \widehat{\boldsymbol{Y}}_{23} - \widehat{\boldsymbol{Y}}_{12}\widehat{\boldsymbol{Y}}_{23} \\ &\leq 2\left(\frac{1}{2} + E\right) - \left(\frac{1}{2} + E\right)^2 \\ &= \frac{3}{4} + E - E^2. \end{aligned}$$

Thus, we obtain $E^2-2E+c-\frac{3}{4}\le 0$ and $(1-E)^2\le \frac{7}{4}-c$. Because $E<1/2$, we have $1-E>0$, and thus $E\ge 1-\sqrt{\frac{7}{4}-c}$. Substituting $c=\sqrt{3}/2$ gives

$$E\ge 1-\sqrt{\frac{7}{4}-\frac{\sqrt{3}}{2}}=1-\frac{1}{2}\sqrt{7-2\sqrt{3}}=E_0.$$

Thus, every exponentially calibrated Euclidean realization has maximum absolute error at least E_0. Finally, we show that this lower bound is attainable. Fix any $\varepsilon>0$, and take three collinear points in $\mathbb{R}$: $v_1=0$, $v_2=t$, and $v_3=2t$, where $t=-\frac{1}{\varepsilon}\log\left(\frac{1}{2}-E_0\right)$. Since $0<E_0<1/2$, this choice is well defined and $t>0$. By construction, we have $e^{-\varepsilon t}=\frac{1}{2}-E_0$, and thus

$$\widehat{\boldsymbol{Y}}_{12}=\widehat{\boldsymbol{Y}}_{23}=1-e^{-\varepsilon t}=\frac{1}{2}+E_0.$$

For the remaining pair, we have

$$\widehat{\boldsymbol{Y}}_{13}=1-e^{-2\varepsilon t}=1-\left(\frac{1}{2}-E_0\right)^2=\frac{3}{4}+E_0-E_0^2.$$

We can easily see that the defining relation $E_0=1-\sqrt{\frac{7}{4}-c}$ is equivalent to $\frac{3}{4}+E_0-E_0^2=c-E_0$. Thus, we get $\widehat{\boldsymbol{Y}}_{13}=c-E_0$. It then follows that

$$|\widehat{\boldsymbol{Y}}_{12}-\boldsymbol{Y}_{12}|=|\widehat{\boldsymbol{Y}}_{23}-\boldsymbol{Y}_{23}|=|\widehat{\boldsymbol{Y}}_{13}-\boldsymbol{Y}_{13}|=E_0.$$

Thus, the lower bound is attained already in one dimension, and $E_\star=E_0$. Since $E_0>0$, exact realization is impossible in every embedding dimension. □

The corollary shows that Euclidean representability of a target does not imply representability after imposing the exponential distance-to-target map.

We now examine whether the graph metanetwork can produce the required embeddings by separating the graph-summary map $\boldsymbol{s}_\psi(G)$ from the shared readout φ_θ and the relation-specific linear head. For a fixed relation, the predictor has the form $\boldsymbol{z}(G)=\varphi_\theta(\boldsymbol{s}_\psi(G))$ and $\widehat{y}(G,G')=q\left(\|\boldsymbol{W}_{\mathsf{h}}\left(\boldsymbol{z}(G)-\boldsymbol{z}(G')\right)\|_2\right)$. We study the expressivity of the graph embedding, in which they identify the restrictions and establish the existence of suitable parameters under explicit assumptions.

Proposition 7. *Fix an integer $R\ge 0$. For each site s, let $G_s=\Psi(\mathcal{C}_s)$ be its finite attributed computation graph, and let $T_s\subseteq V_s$ be the nonempty set of nodes pooled for the target representation site. Consider the R-round independent graph metanetwork in Sec. 3, with shared node initialization, gated messages*

$$\boldsymbol{m}_{ij,s}^{(r)}=\boldsymbol{W}\boldsymbol{h}_{i,s}^{(r)}\odot\psi_{\mathrm{edge}}(\boldsymbol{a}_{ij,s}),\tag{43}$$

and shared residual node updates using both the sum and the mean of incoming messages, followed by nodewise layer normalization. Assume deterministic evaluation and a fixed convention for the mean of an empty incoming neighborhood. Assume also that the global and target pooling operations are shared functions only of their respective multisets of final node states, and that the readout uses only the resulting summary.

For every node $i\in V_s$, let $c_{i,s}^{(0)}$ be a symbolic record of all node attributes used by the encoder, including node type and any target-membership marker supplied as an input. Define

$$c_{j,s}^{(r+1)}=\left(c_{j,s}^{(r)},\{\{(\boldsymbol{a}_{ij,s},c_{i,s}^{(r)}):i\in\mathcal{N}_s^-(j)\}\}\right)\ \text{for } r=0,\dots,R-1,$$

where $\boldsymbol{a}_{ij,s}$ contains all edge attributes used by the message function, $\{\{\cdot\}\}$ denotes a multiset, and symbolic labels are compared across graphs. Suppose two sites u,u' satisfy

$$\{\{c_{i,u}^{(R)}:i\in V_u\}\}=\{\{c_{i,u'}^{(R)}:i\in V_{u'}\}\},$$
$$\{\{c_{i,u}^{(R)}:i\in T_u\}\}=\{\{c_{i,u'}^{(R)}:i\in T_{u'}\}\}.$$

Then, for every common choice of the encoder and readout parameters, we have $\boldsymbol{z}_u=\boldsymbol{z}_{u'}$. For the independent estimator $d_{\phi,\mathsf{h}}(s,t)=\|\boldsymbol{W}_{\mathsf{h}}(\boldsymbol{z}_s-\boldsymbol{z}_t)\|_2$ and $\widehat{y}_{st}^{(\mathsf{h})}=q_{\mathsf{h}}\left(d_{\phi,\mathsf{h}}(s,t)\right)$, with shared relation-specific heads and calibrations, every relation $\mathsf{h}\in\mathcal{H}$ and comparison partner v satisfy

$d_{\phi,\mathsf{h}}(u,v) = d_{\phi,\mathsf{h}}(u',v)$ *and* $\widehat{y}^{(\mathsf{h})}_{uv} = \widehat{y}^{(\mathsf{h})}_{u'v}$. *In addition, we have* $d_{\phi,\mathsf{h}}(u,u') = 0$ *and* $\widehat{y}^{(\mathsf{h})}_{uu'} = q_{\mathsf{h}}(0)$. *If* $\Delta = \left|y^{(\mathsf{h})}_{uv} - y^{(\mathsf{h})}_{u'v}\right| > 0$, *then we also have*

$$\max\left\{\left|\widehat{y}^{(\mathsf{h})}_{uv} - y^{(\mathsf{h})}_{uv}\right|, \left|\widehat{y}^{(\mathsf{h})}_{u'v} - y^{(\mathsf{h})}_{u'v}\right|\right\} \geq \frac{\Delta}{2}.$$

Proof. Fix the shared encoder and readout parameters. We show by induction that, at every round r, the hidden state is a common function of the corresponding symbolic label, i.e., $\boldsymbol{h}^{(r)}_{i,s} = f_r(c^{(r)}_{i,s})$ for $s \in \{u, u'\}$. The statement holds at initialization because the initial symbolic label records all inputs to the shared node initializer.

Suppose it holds at round r. If two receiving nodes have equal round-$(r+1)$ symbolic labels, then they have equal round-r labels and equal incoming multisets of edge attributes and predecessor labels. The induction hypothesis then gives equal receiving-node states and equal incoming multisets of edge attributes and predecessor states.

Since $\boldsymbol{W}$ and ψ_{edge} are shared, the corresponding multisets of gated messages agree. Their sums agree, and their means agree because multiset equality also preserves cardinality. For empty incoming neighborhoods, equality follows from the fixed convention. Thus the inputs to the shared residual update agree, and the shared nodewise layer normalization produces equal next states. This establishes the common function f_{r+1} and completes the induction.

Applying f_R to the two assumed multiset equalities gives equal multisets of final hidden states over the complete graphs and over the target pooling sets. The shared pooling operations then produce equal summaries, and the shared readout gives $\boldsymbol{z}_u = \boldsymbol{z}_{u'}$. For every relation h and partner v, it follows that $\boldsymbol{W}_{\mathsf{h}}(\boldsymbol{z}_u - \boldsymbol{z}_v) = \boldsymbol{W}_{\mathsf{h}}(\boldsymbol{z}_{u'} - \boldsymbol{z}_v)$. Taking Euclidean norms and applying the same calibration q_{h} proves equality of the latent distances and predicted targets. Likewise, $\boldsymbol{z}_u - \boldsymbol{z}_{u'} = \boldsymbol{0}$ gives $d_{\phi,\mathsf{h}}(u,u') = 0$ and $\widehat{y}^{(\mathsf{h})}_{uu'} = q_{\mathsf{h}}(0)$. Finally, let $\alpha = \widehat{y}^{(\mathsf{h})}_{uv} = \widehat{y}^{(\mathsf{h})}_{u'v}$ for the common prediction. The triangle inequality gives

$$\Delta \leq \left|y^{(\mathsf{h})}_{uv} - \alpha\right| + \left|\alpha - y^{(\mathsf{h})}_{u'v}\right| \leq 2\max\left\{\left|\widehat{y}^{(\mathsf{h})}_{uv} - y^{(\mathsf{h})}_{uv}\right|, \left|\widehat{y}^{(\mathsf{h})}_{u'v} - y^{(\mathsf{h})}_{u'v}\right|\right\}.$$

□

We next assume that one common attainable backbone separates a finite bank, allowing us to isolate the constraints imposed by the calibrated Euclidean head from those imposed by graphs.

Theorem 8 (Finite-bank realizability and spectral approximation)**.** *Fix graphs* $G_1, \ldots, G_N$, *with* $N \geq 2$, *and an attainable setting* ψ *of the shared backbone and pooling maps such that*

$$\boldsymbol{s}_i = \boldsymbol{s}_\psi(G_i) \in \mathbb{R}^p, \text{ for } i = 1, \ldots, N, \tag{44}$$

are pairwise distinct. Keep this backbone setting fixed. Consider one relation, with embedding dimension $k \geq 1$ *and head dimension* $k_{\mathsf{h}} \geq 1$, *and predictors:*

$$\boldsymbol{z}_i = \rho(\boldsymbol{s}_i) \in \mathbb{R}^k, \text{ and } \widehat{\boldsymbol{Y}}_{ij} = q\left(\|\boldsymbol{W}_{\mathsf{h}}(\boldsymbol{z}_i - \boldsymbol{z}_j)\|_2\right), \tag{45}$$

where $\boldsymbol{W}_{\mathsf{h}} \in \mathbb{R}^{k_{\mathsf{h}} \times k}$ *is unrestricted. Allow the shared readout* ρ *to range over finite-width ReLU MLPs with biases and an affine output, with unrestricted hidden widths and weights. Let* $q : [0,\infty) \to [0,\infty)$ *be a fixed strictly increasing function satisfying* $q(0) = 0$. *Let* $\boldsymbol{Y} = (\boldsymbol{Y}_{ij}) \in \mathbb{R}^{N \times N}$ *be symmetric, entrywise nonnegative, and zero-diagonal, with every entry in* $q([0,\infty))$. *Define* $\boldsymbol{D}_{ij} = q^{-1}(Y_{ij})$, $\boldsymbol{D} = (\boldsymbol{D}_{ij})$, $d_* = \min\{k, k_{\mathsf{h}}\}$, $\boldsymbol{J}_N = \boldsymbol{I}_N - \frac{1}{N}\boldsymbol{1}_N\boldsymbol{1}_N^\top$, *and* $\boldsymbol{B}_{\mathrm{D}} = -\frac{1}{2}\boldsymbol{J}_N\boldsymbol{D}^{\circ 2}\boldsymbol{J}_N$, *where* $\boldsymbol{D}^{\circ 2}$ *denotes the entrywise square. Then, there exists a shared readout* ρ *and a head* $\boldsymbol{W}_{\mathsf{h}}$ *satisfying* $\widehat{\boldsymbol{Y}} = \boldsymbol{Y}$ *on the entire bank if and only if* $\boldsymbol{B}_{\mathrm{D}} \succeq 0$ *and* $\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) \leq d_*$. *When these conditions hold, a readout with one hidden ReLU layer of width at most* $N - 1$ *suffices.*

In addition, suppose $\boldsymbol{B}_{\mathrm{D}} \succeq 0$, *and denote its eigenvalues as* $\lambda_1 \geq \cdots \geq \lambda_N \geq 0$. *Let* $d_{\max} = \max_{i,j} D_{ij}$. *If* q *is* L_q*-Lipschitz on* $[0, d_{\max}]$, *then for every integer* $1 \leq r \leq \min\{d_*, N-1\}$, *there is an attainable predictor* $\widehat{\boldsymbol{Y}}^{(r)}$ *such that* $0 \leq \widehat{\boldsymbol{Y}}^{(r)}_{ij} \leq \boldsymbol{Y}_{ij}$ *for all* i, j, *and*

$$\max_{i,j}\left|\widehat{\boldsymbol{Y}}^{(r)}_{ij} - Y_{ij}\right| \leq L_q\sqrt{2\lambda_{r+1}}. \tag{46}$$

The bound is zero when $r \geq \operatorname{rank}(\boldsymbol{B}_{\mathrm{D}})$.

Proof. Suppose the predictor realizes $\boldsymbol{Y}$. Define $\boldsymbol{e}_i = \boldsymbol{W}_{\mathsf{h}}\boldsymbol{z}_i$ and $\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1^\top \\ \vdots \\ \boldsymbol{e}_N^\top \end{bmatrix}$. Since q is injective, we have $\boldsymbol{D}_{ij} = \|\boldsymbol{e}_i - \boldsymbol{e}_j\|_2$. Let $\boldsymbol{\nu} \in \mathbb{R}^N$ have entries $\nu_i = \|\boldsymbol{e}_i\|_2^2$. Then, we have

$$\boldsymbol{D}^{\circ 2} = \boldsymbol{\nu}\mathbf{1}_N^\top + \mathbf{1}_N\boldsymbol{\nu}^\top - 2\boldsymbol{E}\boldsymbol{E}^\top.$$

Using $\boldsymbol{J}_N\mathbf{1}_N = \mathbf{0}$, we can see that

$$\boldsymbol{B}_{\mathrm{D}} = (\boldsymbol{J}_N\boldsymbol{E})(\boldsymbol{J}_N\boldsymbol{E})^\top \succeq 0.$$

Moreover, if $\boldsymbol{z}$ has rows $\boldsymbol{z}_i^\top$, then $\boldsymbol{E} = \boldsymbol{z}\boldsymbol{W}_{\mathsf{h}}^\top$, and thus

$$\operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) \le \operatorname{rank}(\boldsymbol{E}) \le \operatorname{rank}(\boldsymbol{W}_{\mathsf{h}}) \le d_*.$$

We now prove in another direction. Suppose $\boldsymbol{B}_{\mathrm{D}} \succeq 0$ and $r_0 = \operatorname{rank}(\boldsymbol{B}_{\mathrm{D}}) \le d_*$. Let $\boldsymbol{b}_i$ denote the i-th standard basis vector in $\mathbb{R}^N$. Since $\boldsymbol{J}_N(\boldsymbol{b}_i - \boldsymbol{b}_j) = \boldsymbol{b}_i - \boldsymbol{b}_j$ and $\boldsymbol{D}$ is symmetric and zero-diagonal, we have

$$(\boldsymbol{b}_i - \boldsymbol{b}_j)^\top \boldsymbol{B}_{\mathrm{D}}(\boldsymbol{b}_i - \boldsymbol{b}_j) = -\frac{1}{2}(\boldsymbol{b}_i - \boldsymbol{b}_j)^\top \boldsymbol{D}^{\circ 2}(\boldsymbol{b}_i - \boldsymbol{b}_j) = \boldsymbol{D}_{ij}^2.$$

If $r_0 = 0$, this identity gives $\boldsymbol{D} = \mathbf{0}$ and thus $\boldsymbol{Y} = \mathbf{0}$. A constant-zero readout realizes the target matrix. Now suppose $r_0 > 0$ and factor $\boldsymbol{B}_{\mathrm{D}} = \boldsymbol{X}\boldsymbol{X}^\top$ with $\boldsymbol{X} \in \mathbb{R}^{N\times r_0}$. Denote the rows of $\boldsymbol{X}$ as $\boldsymbol{x}_i^\top$. Then, we have $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 = \boldsymbol{D}_{ij}^2$ and $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \boldsymbol{D}_{ij}$. Also, $\boldsymbol{B}_{\mathrm{D}}\mathbf{1}_N = \mathbf{0}$ implies $\boldsymbol{X}^\top\mathbf{1}_N = \mathbf{0}$, so these coordinates are centered. Since $\boldsymbol{s}_i$ are pairwise distinct, there exists $\boldsymbol{a} \in \mathbb{R}^p$ such that $t_i = \boldsymbol{a}^\top\boldsymbol{s}_i$ are pairwise distinct. Note that the forbidden choices of $\boldsymbol{a}$ form a finite union of proper hyperplanes. After relabeling the indices and assuming $t_1 < \cdots < t_N$. Define $\boldsymbol{v}_0 = \mathbf{0}$, $\boldsymbol{v}_j = \frac{\boldsymbol{x}_{j+1} - \boldsymbol{x}_j}{t_{j+1} - t_j}$ for $j = 1, \ldots, N-1$, and the vector-valued piecewise-linear function

$$f(t) = \boldsymbol{x}_1 + \sum_{j=1}^{N-1} (\boldsymbol{v}_j - \boldsymbol{v}_{j-1})(t - t_j)_+,$$

where $(t - t_j)_+ = \max\{0, t - t_j\}$. This function satisfies $f(t_1) = \boldsymbol{x}_1$ and has slope $\boldsymbol{v}_j$ on (t_j, t_{j+1}). Thus, we have $f(t_i) = \boldsymbol{x}_i$ for every i. The map $\boldsymbol{s} \mapsto f(\boldsymbol{a}^\top\boldsymbol{s})$ is a ReLU network with one hidden layer of width $N-1$ and an affine output. Pad this output with zeros to obtain a readout $\rho : \mathbb{R}^p \to \mathbb{R}^k$ satisfying $\rho(\boldsymbol{s}_i) = \begin{bmatrix} \boldsymbol{x}_i \\ \mathbf{0} \end{bmatrix}$. Choose $\boldsymbol{W}_{\mathsf{h}}$ to copy the first r_0 coordinates into $\mathbb{R}^{k_{\mathsf{h}}}$ and set all remaining output coordinates to zero. Then, we have $\|\boldsymbol{W}_{\mathsf{h}}(\rho(\boldsymbol{s}_i) - \rho(\boldsymbol{s}_j))\|_2 = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \boldsymbol{D}_{ij}$, and applying q, we get $\widehat{\boldsymbol{Y}} = \boldsymbol{Y}$.

For the spectral approximation, let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ be an orthonormal eigenbasis of $\boldsymbol{B}_{\mathrm{D}}$ corresponding to the ordered eigenvalues, and define

$$\boldsymbol{B}_r = \sum_{\ell=1}^{r} \lambda_\ell \boldsymbol{u}_\ell \boldsymbol{u}_\ell^\top, \qquad \boldsymbol{C}_r = \boldsymbol{B}_{\mathrm{D}} - \boldsymbol{B}_r.$$

Then, we can see that $\boldsymbol{C}_r \succeq 0$ and $\|\boldsymbol{C}_r\|_{\mathrm{op}} = \lambda_{r+1}$. Factor $\boldsymbol{B}_r = \boldsymbol{X}_r\boldsymbol{X}_r^\top$ with $\boldsymbol{X}_r \in \mathbb{R}^{N\times r}$, and let $\boldsymbol{D}_{ij}^{(r)}$ be the Euclidean distances between its rows. For every i, j, we have

$$\begin{aligned} 0 &\le \boldsymbol{D}_{ij}^2 - \left(\boldsymbol{D}_{ij}^{(r)}\right)^2 \\ &= (\boldsymbol{b}_i - \boldsymbol{b}_j)^\top \boldsymbol{C}_r (\boldsymbol{b}_i - \boldsymbol{b}_j) \\ &\le \|\boldsymbol{C}_r\|_{\mathrm{op}} \|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2^2 \\ &\le 2\lambda_{r+1}. \end{aligned}$$

Thus $0 \le \boldsymbol{D}_{ij}^{(r)} \le \boldsymbol{D}_{ij}$. Using $(x-y)^2 \le x^2 - y^2$ for $x \ge y \ge 0$, we obtain

$$0 \le \boldsymbol{D}_{ij} - \boldsymbol{D}_{ij}^{(r)} \le \sqrt{2\lambda_{r+1}}.$$

The interpolation construction above realizes the rows of $\boldsymbol{X}_r$ through the readout and head, since $r \le d_*$. For the resulting predictor, we have $\widehat{\boldsymbol{Y}}_{ij}^{(r)} = q(\boldsymbol{D}_{ij}^{(r)})$. Both $\boldsymbol{D}_{ij}^{(r)}$ and $\boldsymbol{D}_{ij}$ lie in $[0, d_{\max}]$.

Monotonicity and the Lipschitz property of q thus give

$$\begin{aligned} 0 &\le \boldsymbol{Y}_{ij} - \widehat{\boldsymbol{Y}}_{ij}^{(r)} \\ &= q(\boldsymbol{D}_{ij}) - q(\boldsymbol{D}_{ij}^{(r)}) \\ &\le L_q \left(\boldsymbol{D}_{ij} - \boldsymbol{D}_{ij}^{(r)} \right) \\ &\le L_q \sqrt{2\lambda_{r+1}}. \end{aligned}$$

□

To study approximation throughout a domain, we next assume compactness and continuity.

Theorem 9 (Uniform approximation)**.** *Let $\mathcal{K}$ be a nonempty compact metric space, and fix a continuous attainable $\boldsymbol{s}_\psi : \mathcal{K} \to \mathbb{R}^p$. Let $\Phi : \mathcal{K} \to \mathcal{H}$ be continuous, where $\mathcal{H}$ is a real Hilbert space. Define $S = \boldsymbol{s}_\psi(\mathcal{K})$, $M_\Phi = \sup_{G \in \mathcal{K}} \|\Phi(G)\|_{\mathcal{H}}$, and $\rho(G, G') = \|\Phi(G) - \Phi(G')\|_{\mathcal{H}}$. Let $q : [0, \infty) \to \mathbb{R}$ be fixed and L_q-Lipschitz on $[0, 2M_\Phi + 2]$. For a target function $y : \mathcal{K} \times \mathcal{K} \to \mathbb{R}$, assume*

$$\kappa = \sup_{G, G' \in \mathcal{K}} |q(\rho(G, G')) - y(G, G')| < \infty. \tag{47}$$

Consider independent predictors

$$\boldsymbol{z}_\theta(G) = \varphi_\theta(\boldsymbol{s}_\psi(G)) \in \mathbb{R}^k \text{ and } \widehat{y}(G, G') = q\left(\|\boldsymbol{W}_{\mathsf{h}} \left(\boldsymbol{z}_\theta(G) - \boldsymbol{z}_\theta(G') \right)\|_2 \right), \tag{48}$$

where $\boldsymbol{W}_{\mathsf{h}} \in \mathbb{R}^{k_{\mathsf{h}} \times k}$ is unrestricted. Assume that the allowed readouts uniformly approximate arbitrary continuous finite-dimensional maps on compact subsets of $\mathbb{R}^p$, with error measured in the Euclidean norm.

For a finite-dimensional subspace $V \subseteq \mathcal{H}$, we define

$$\alpha_{\psi, V} = \sup_{\boldsymbol{s} \in S} \inf_{\mathbf{c} \in V} \sup_{\substack{G \in \mathcal{K} \\ \boldsymbol{s}_\psi(G) = \boldsymbol{s}}} \|P_V \Phi(G) - \mathbf{c}\|_{\mathcal{H}}, \tag{49}$$

and

$$\boldsymbol{r}_V(G) = \Phi(G) - P_V \Phi(G), \qquad \beta_V = \sup_{G, G' \in \mathcal{K}} \|\boldsymbol{r}_V(G) - \boldsymbol{r}_V(G')\|_{\mathcal{H}}. \tag{50}$$

Then, for every $0 < a \le 1$ and positive integers k, k_{h} satisfying $k \ge \dim V$ and $k_{\mathsf{h}} \ge \dim V$, there exists a shared readout and a linear head such that

$$\sup_{G, G' \in \mathcal{K}} |\widehat{y}(G, G') - y(G, G')| \le L_q \sqrt{4(\alpha_{\psi, V} + a)^2 + \beta_V^2} + \kappa. \tag{51}$$

Proof. Fix V and a, and abbreviate $\alpha = \alpha_{\psi, V}$. For each $\boldsymbol{s} \in S$, the set

$$\mathcal{A}_{\boldsymbol{s}, V} = \{P_V \Phi(G) : G \in \mathcal{K},\ \boldsymbol{s}_\psi(G) = \boldsymbol{s}\}$$

is nonempty and compact. Since V is finite-dimensional, there is a center $\mathbf{c}_{\boldsymbol{s}} \in V$ minimizing

$$\sup_{\mathbf{v} \in \mathcal{A}_{\boldsymbol{s}, V}} \|\mathbf{v} - \mathbf{c}_{\boldsymbol{s}}\|_{\mathcal{H}}.$$

A minimizing center can be chosen in the closed convex hull of $\mathcal{A}_{\boldsymbol{s}, V}$, i.e., projecting any center onto that convex hull cannot increase its distance to any point of $\mathcal{A}_{\boldsymbol{s}, V}$. Therefore, we have $\|\mathbf{c}_{\boldsymbol{s}}\|_{\mathcal{H}} \le M_\Phi$ and and

$$\sup_{\substack{G \in \mathcal{K} \\ \boldsymbol{s}_\psi(G) = \boldsymbol{s}}} \|P_V \Phi(G) - \mathbf{c}_{\boldsymbol{s}}\|_{\mathcal{H}} \le \alpha.$$

Now, we define the compact set

$$\mathcal{C}_{\boldsymbol{s}} = \left\{ G \in \mathcal{K} : \|P_V \Phi(G) - \mathbf{c}_{\boldsymbol{s}}\|_{\mathcal{H}} \ge \alpha + \frac{a}{2} \right\}.$$

The set $\boldsymbol{s}_\psi(\mathcal{C}_{\boldsymbol{s}})$ is compact and does not contain $\boldsymbol{s}$. Thus, $\mathcal{U}_{\boldsymbol{s}} = S \setminus \boldsymbol{s}_\psi(\mathcal{C}_{\boldsymbol{s}})$ is an open neighborhood of $\boldsymbol{s}$. For every G whose summary lies in $\mathcal{U}_{\boldsymbol{s}}$, we have $\|P_V \Phi(G) - \mathbf{c}_{\boldsymbol{s}}\|_{\mathcal{H}} < \alpha + \frac{a}{2}$. Choose a finite subcover $U_{\boldsymbol{s}_1}, \dots, U_{\boldsymbol{s}_m}$ of S and continuous nonnegative functions $\chi_1, \dots, \chi_m$ such that

$$\sum_{j=1}^{m} \chi_j(\boldsymbol{s}) = 1, \qquad \chi_j(\boldsymbol{s}) > 0 \implies \boldsymbol{s} \in U_{\boldsymbol{s}_j}.$$

Such functions can be constructed by normalizing $w_j(\boldsymbol{s}) = \operatorname{dist}\left(\boldsymbol{s}, S \setminus U_{\boldsymbol{s}_j}\right)$, with $w_j \equiv 1$ when $U_{\boldsymbol{s}_j} = S$. Their sum is positive everywhere because the sets form an open cover. The continuous map $F_V(\boldsymbol{s}) = \sum_{j=1}^{m} \chi_j(\boldsymbol{s}) \mathbf{c}_{\boldsymbol{s}_j} \in V$ then satisfies $\sup_{\boldsymbol{s}\in S} \|F_V(\boldsymbol{s})\|_{\mathcal{H}} \le M_\Phi$ and $\sup_{G\in\mathcal{K}} \|F_V(\boldsymbol{s}_\psi(G)) - P_V\Phi(G)\|_{\mathcal{H}} \le \alpha + \frac{a}{2}$.

Choose an orthonormal basis of V, identify F_V with its coordinate map into $\mathbb{R}^{\dim V}$, and pad with zeros to dimension k. The readout approximation property gives a shared readout with uniform Euclidean error at most $a/2$ from this padded map. Choose $\boldsymbol{W}_{\mathsf{h}}$ to select these coordinates and pad with zeros in the head output. Identifying the selected coordinates with a vector $\boldsymbol{w}(G) \in V$, we obtain

$$\sup_{G\in\mathcal{K}} \|\boldsymbol{w}(G) - F_V(\boldsymbol{s}_\psi(G))\|_{\mathcal{H}} \le \frac{a}{2}.$$

Therefore, we have

$$\sup_{G\in\mathcal{K}} \|\boldsymbol{w}(G) - P_V\Phi(G)\|_{\mathcal{H}} \le \alpha + a, \text{and} \sup_{G\in\mathcal{K}} \|\boldsymbol{w}(G)\|_{\mathcal{H}} \le M_\Phi + \frac{a}{2}.$$

The distance used by the predictor is then

$$\widehat{\rho}(G, G') = \|\boldsymbol{w}(G) - \boldsymbol{w}(G')\|_{\mathcal{H}}.$$

If $V = \{0\}$, take a zero head and interpret these coordinate maps as empty.

Let $\boldsymbol{u}(G) = \boldsymbol{w}(G) - P_V\Phi(G) \in V$. For any $G, G' \in \mathcal{K}$, the difference between the predicted and ideal difference vectors is

$$(\boldsymbol{w}(G) - \boldsymbol{w}(G')) - (\Phi(G) - \Phi(G')) = (\boldsymbol{u}(G) - \boldsymbol{u}(G')) - (\boldsymbol{r}_V(G) - \boldsymbol{r}_V(G')).$$

The first term lies in V and the second lies in $V^\perp$. Thus the reverse triangle inequality and orthogonality give

$$\begin{aligned} |\widehat{\rho}(G, G') - \rho(G, G')| &\le \left(\|\boldsymbol{u}(G) - \boldsymbol{u}(G')\|_{\mathcal{H}}^2 + \|\boldsymbol{r}_V(G) - \boldsymbol{r}_V(G')\|_{\mathcal{H}}^2\right)^{1/2} \\ &\le \sqrt{4(\alpha + a)^2 + \beta_V^2}. \end{aligned}$$

In addition, we also have $\rho(G, G') \le 2M_\Phi$ and $\widehat{\rho}(G, G') \le 2M_\Phi + a$, thus both distances lie in the stated Lipschitz interval. Therefore, we have

$$\begin{aligned} |\widehat{y}(G, G') - y(G, G')| &\le |q(\widehat{\rho}(G, G')) - q(\rho(G, G'))| + |q(\rho(G, G')) - y(G, G')| \\ &\le L_q\sqrt{4(\alpha + a)^2 + \beta_V^2} + \kappa. \end{aligned}$$

□

C.3 Score error, layer retrieval and stitching

Proposition 10 (Stability of retrieval and Hausdorff aggregation)**.** *Let $\mathcal{X}$ be a set of sites, let $y, \widehat{y} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and let $E \ge 0$.*

(i) Retrieval. *Fix a query $u \in \mathcal{X}$ and a nonempty finite candidate set $\mathcal{V} \subseteq \mathcal{X}$, and suppose that $|\widehat{y}_{uv} - y_{uv}| \le E$ for every $v \in \mathcal{V}$. Choose, with arbitrary tie-breaking, $v^\star \in \arg\min_{v\in\mathcal{V}} y_{uv}$ and $\widehat{v} \in \arg\min_{v\in\mathcal{V}} \widehat{y}_{uv}$. Then $0 \le y_{u\widehat{v}} - y_{uv^\star} \le 2E$. Denote $n = |\mathcal{V}|$, and let $y_{u,(1)} \le \cdots \le y_{u,(n)}$ be the target scores in nondecreasing order, counted with multiplicity. For $1 \le k < n$, if $y_{u,(k+1)} - y_{u,(k)} > 2E$, then $T_k = \{v \in \mathcal{V} : y_{uv} \le y_{u,(k)}\}$ has cardinality k, and every selection of the k smallest predicted scores recovers T_k, regardless of tie-breaking. Specifically, if $n \ge 2$ and $y_{u,(2)} - y_{u,(1)} > 2E$, then $v^\star$ is the unique minimizer of both the target and predicted scores, so $\widehat{v} = v^\star$. For $k = n$, top-k recovery is obtained and the nearest-site recovery is also obtained when $n = 1$.*

(ii) Hausdorff aggregation. *For nonempty finite sets $A, B \subseteq \mathcal{X}$ and $z \in \{y, \widehat{y}\}$, define*

$$\mathfrak{H}_z(A, B) = \max\left\{\max_{a\in A}\min_{b\in B} z_{ab}, \max_{b\in B}\min_{a\in A} z_{ab}\right\}. \tag{52}$$

If $|\widehat{y}_{ab} - y_{ab}| \le E$ for every $(a, b) \in A \times B$, then $|\mathfrak{H}_{\widehat{y}}(A, B) - \mathfrak{H}_y(A, B)| \le E$. If y is a pseudometric on $\mathcal{X}$, then $\mathfrak{H}_y$ is a pseudometric on the collection of nonempty finite subsets of $\mathcal{X}$. If y is a metric, then $\mathfrak{H}_y$ is a metric on that collection.

Proof. For the retrieval bound, the optimality of $\widehat{v}$ for the predicted scores gives

$$y_{u\widehat{v}} \le \widehat{y}_{u\widehat{v}} + E \le \widehat{y}_{uv^\star} + E \le y_{uv^\star} + 2E.$$

The lower bound $y_{u\widehat{v}} - y_{uv^\star} \ge 0$ follows from optimality of $v^\star$ for the target scores. Now suppose that $1 \le k < n$ and $y_{u,(k+1)} - y_{u,(k)} > 2E$. Specifically, $y_{u,(k)} < y_{u,(k+1)}$, so T_k has k elements. For every $v \in T_k$ and $w \in \mathcal{V} \setminus T_k$, we have

$$\widehat{y}_{uv} \le y_{u,(k)} + E < y_{u,(k+1)} - E \le \widehat{y}_{uw}.$$

Thus every candidate in T_k has a strictly smaller predicted score than every candidate outside T_k. Therefore, every predicted top-k selection equals T_k. The nearest-site statement follows by taking $k = 1$. The cases $k = n$ and $n = 1$ are then obtained.

For the aggregation bound, the entrywise error assumption implies, for each $a \in A$, we have

$$\min_{b\in B} y_{ab} - E \le \min_{b\in B} \widehat{y}_{ab} \le \min_{b\in B} y_{ab} + E.$$

Taking the maximum over $a \in A$ preserves these inequalities. The same argument applies to the column minima and their maximum over $b \in B$. Taking the maximum of the two resulting terms then gives

$$\mathfrak{H}_y(A, B) - E \le \mathfrak{H}_{\widehat{y}}(A, B) \le \mathfrak{H}_y(A, B) + E.$$

Finally, suppose that y is a pseudometric, and define

$$h_y(A, B) = \max_{a\in A} \min_{b\in B} y_{ab}.$$

By symmetry of y, we have

$$\mathfrak{H}_y(A, B) = \max\{h_y(A, B), h_y(B, A)\}.$$

Let A, B, C be nonempty finite subsets of $\mathcal{X}$. For each $a \in A$, choose $b_a \in \arg\min_{b\in B} y_{ab}$ and $c_a \in \arg\min_{c\in C} y_{b_a c}$. These minimizers exist since the sets are nonempty and finite. The triangle inequality for y then gives

$$\min_{c\in C} y_{ac} \le y_{ac_a} \le y_{ab_a} + y_{b_a c_a} \le h_y(A, B) + h_y(B, C).$$

Taking the maximum over $a \in A$ shows that

$$h_y(A, C) \le h_y(A, B) + h_y(B, C).$$

Applying the same argument in the reverse direction gives

$$h_y(C, A) \le h_y(C, B) + h_y(B, A).$$

Thus, we have

$$\begin{aligned}\mathfrak{H}_y(A, C) &= \max\{h_y(A, C), h_y(C, A)\} \\ &\le \max\{h_y(A, B) + h_y(B, C), h_y(C, B) + h_y(B, A)\} \\ &\le \mathfrak{H}_y(A, B) + \mathfrak{H}_y(B, C),\end{aligned}$$

and we also have nonnegativity and symmetry. In addition, note that $\mathfrak{H}_y(A, A) = 0$ because $y_{aa} = 0$ for every $a \in A$. Thus $\mathfrak{H}_y$ is a pseudometric. If y is a metric and $\mathfrak{H}_y(A, B) = 0$, then every $a \in A$ has some $b \in B$ with $y_{ab} = 0$, and thus $a = b$. Thus $A \subseteq B$. The reverse inclusion follows similarly. Thus, $A = B$ and $\mathfrak{H}_y$ is a metric. □

Proposition 11. *Let a, b index two supervised pairs, with real-valued targets y_a, y_b and predictions $\widehat{y}_a, \widehat{y}_b$. Fix $\tau > 0$, use the convention $\operatorname{sign}(0) = 0$, and define $s_{ab} = \operatorname{sign}(y_a - y_b)$, $m_{ab} = s_{ab}(\widehat{y}_a - \widehat{y}_b)$, and $\ell_\tau(a, b) = \log(1 + \exp(-\tau m_{ab}))$. Then, we have*

$$\mathbf{1}\{y_a \ne y_b,\ m_{ab} \le 0\} \le \frac{\mathbf{1}\{y_a \ne y_b\}\,\ell_\tau(a, b)}{\log 2} \le \frac{\ell_\tau(a, b)}{\log 2}. \tag{53}$$

Thus, under any joint distribution of the compared targets and predictions, the ranking error probability $R_{\mathrm{rank}} := \mathbb{P}(y_a \ne y_b,\ m_{ab} \le 0)$ satisfies

$$R_{\mathrm{rank}} \le \frac{\mathbb{E}[\mathbf{1}\{y_a \ne y_b\}\,\ell_\tau(a, b)]}{\log 2} = \frac{\mathbb{E}[\ell_\tau(a, b)]}{\log 2} - \mathbb{P}(y_a = y_b) \le \frac{\mathbb{E}[\ell_\tau(a, b)]}{\log 2}. \tag{54}$$

The reversed predictions and predicted ties are counted as errors only when the targets are unequal and the multiplicative constant $1/\log 2$ is optimal.

Proof. Note that if $y_a = y_b$, then the error indicator and the masked loss $\mathbf{1}\{y_a \neq y_b\}\ell_\tau(a,b)$ both vanish. Suppose now that $y_a \neq y_b$. If $m_{ab} \leq 0$, then $\tau > 0$ implies

$$\ell_\tau(a,b) = \log\left(1 + \exp(-\tau m_{ab})\right) \geq \log(1 + \exp(0)) = \log 2.$$

If $m_{ab} > 0$, the error indicator is zero, while the loss is nonnegative.

When $y_a = y_b$, we have $s_{ab} = 0$ and thus $\ell_\tau(a,b) = \log 2$. Thus, we have

$$\ell_\tau(a,b) = \mathbf{1}\{y_a \neq y_b\}\,\ell_\tau(a,b) + \mathbf{1}\{y_a = y_b\}\,\log 2.$$

The final inequality follows from $\mathbb{P}(y_a = y_b) \geq 0$. If the expected loss is infinite, the bounds hold in the extended-real sense. Finally, take $y_a = 1$, $y_b = 0$, and $\widehat{y}_a = \widehat{y}_b = 0$. Then the error indicator equals 1 and $\ell_\tau(a,b) = \log 2$ for every $\tau > 0$. Thus any universal constant C satisfying

$$\mathbf{1}\{y_a \neq y_b,\ m_{ab} \leq 0\} \leq C\,\ell_\tau(a,b)$$

need to satisfy $C \geq 1/\log 2$. □

We fix one relation h and a finite collection of supervised site pairs $\mathcal{P} = \{p_1, \ldots, p_n\}$, with $n \geq 2$. Denote $y_i = y_{p_i}^{(\mathsf{h})}$, $\widehat{y}_i = \widehat{y}_{p_i}^{(\mathsf{h})}$, and $\varepsilon_i = \widehat{y}_i - y_i$, and suppress the relation superscript below.

If an index I is chosen uniformly from $\{1, \ldots, n\}$, then $Y = y_I$ and $\widehat{Y} = \widehat{y}_I$ are the target and prediction on the same random site pair. For $\beta > 0$, we define

$$h_\beta(t) = \begin{cases} t^2/(2\beta), & |t| \leq \beta, \\ |t| - \beta/2, & |t| > \beta, \end{cases}$$

and the empirical regression loss

$$\mathcal{L}_{\mathrm{reg}} = \mathbb{E}[h_\beta(\widehat{Y} - Y)] = \frac{1}{n}\sum_{i=1}^{n} h_\beta(\varepsilon_i).$$

For $\tau > 0$, let $\mathrm{sign}(0) = 0$ and define $s_{ij} = \mathrm{sign}(y_i - y_j)$ and $\ell_{ij} = \log\left(1 + \exp[-\tau s_{ij}(\widehat{y}_i - \widehat{y}_j)]\right)$. For the analysis, we use the complete-comparison ordinal average $M = \binom{n}{2}$ and $\mathcal{L}_{\mathrm{rank}} = \frac{1}{M}\sum_{i<j}\ell_{ij}$. Therefore, the comparisons are averaged uniformly over distinct examples, and the target ties are retained.

For $t \geq 0$, we define the fraction of comparisons with target gap at most t and their contribution to the ordinal loss by

$$F_n(t) = \frac{1}{M}\sum_{i<j}\mathbf{1}\{|y_i - y_j| \leq t\} \quad \text{and} \quad \mathcal{L}_{\mathrm{rank}}^{\leq t} = \frac{1}{M}\sum_{i<j}\mathbf{1}\{|y_i - y_j| \leq t\}\ell_{ij}.$$

Both quantities are normalized by all M comparisons.

Proposition 12 (Joint loss control of empirical Spearman correlation)**.** *Suppose that $|\widehat{Y} - Y| \leq B$ almost surely for some $0 \leq B < \infty$, equivalently $|\varepsilon_i| \leq B$ for every i. Assume that $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ $\widehat{\boldsymbol{y}} = (\widehat{y}_1, \ldots, \widehat{y}_n)^\top$ are not constant. Let $\boldsymbol{r}$ and $\widehat{\boldsymbol{r}}$ be their increasing rank vectors, assigning average ranks to tied values. Then, we have*

$$1 - \rho_{\mathrm{S}} \leq \Gamma_n \inf_{t>0}\left[\min\left\{F_n(t), \frac{\mathcal{L}_{\mathrm{rank}}^{\leq t}}{\log 2}\right\} + \frac{2n\,C_\beta(B)}{(n-1)t^2}\mathcal{L}_{\mathrm{reg}}\right], \tag{55}$$

where $\bar{r} = \frac{n+1}{2}$, $A = \|\boldsymbol{r} - \bar{r}\mathbf{1}_n\|_2$, $\widehat{A} = \|\widehat{\boldsymbol{r}} - \bar{r}\mathbf{1}_n\|_2$, $\rho_{\mathrm{S}} = \frac{(\boldsymbol{r}-\bar{r}\mathbf{1}_n)^\top(\widehat{\boldsymbol{r}}-\bar{r}\mathbf{1}_n)}{A\widehat{A}}$, $\Gamma_n = \frac{nM}{2A\widehat{A}}$, and $C_\beta(B) = 2\max\{B, \beta\}$. If neither vector contains ties, then $\Gamma_n = \frac{3n}{n+1}$. If in addition $Y, \widehat{Y} \in [0,1]$ almost surely and $\beta = 1$, we can take $B = 1$ and $C_1(1) = 2$ and in this case, we have $\mathbb{E}[(\widehat{Y} - Y)^2] = 2\mathcal{L}_{\mathrm{reg}}$.

Proof. Note that for $|x| \leq \beta$, we have $x^2 = 2\beta h_\beta(x)$. For $\beta < |x| \leq B$, we have $h_\beta(x) = |x| - \frac{\beta}{2} \geq \frac{|x|}{2}$ and $x^2 \leq 2Bh_\beta(x)$. Thus, we have $r_n \leq C_\beta(B)\mathcal{L}_{\mathrm{reg}}$ and $\frac{1}{n}\sum_{i=1}^n |\varepsilon_i| \leq \sqrt{r_n}$, where $r_n = \frac{1}{n}\sum_i \varepsilon_i^2$ and the second inequality is Cauchy-Schwarz. If $B \leq \beta$, every error lies in the quadratic branch and thus $r_n = 2\beta\mathcal{L}_{\mathrm{reg}}$.

Let $\widehat{s}_{ij} = \mathrm{sign}(\widehat{y}_i - \widehat{y}_j)$, $\xi_{ij} = \frac{s_{ij} - \widehat{s}_{ij}}{2}$, and $Q_n = \sum_{i<j}\xi_{ij}^2$. Average ranks satisfy $r_i = \bar{r} + \frac{1}{2}\sum_{j=1}^n s_{ij}$ and $\widehat{r}_i = \bar{r} + \frac{1}{2}\sum_{j=1}^n \widehat{s}_{ij}$. Thus, for $\boldsymbol{d} = \boldsymbol{r} - \widehat{\boldsymbol{r}}$, we have $d_i = \sum_j \xi_{ij}$, $\xi_{ji} = -\xi_{ij}$, and

$\sum_i d_i = 0$. By antisymmetry and Cauchy-Schwarz, we obtain

$$\|\boldsymbol{d}\|_2^2 = \sum_{i<j}(d_i - d_j)\xi_{ij} \le \left(\sum_{i<j}(d_i - d_j)^2\right)^{1/2} Q_n^{1/2} = \sqrt{n}\,\|\boldsymbol{d}\|_2 Q_n^{1/2},$$

where the last equality uses $\sum_{i<j}(d_i - d_j)^2 = n\|\boldsymbol{d}\|_2^2$ when $\sum_i d_i = 0$. Dividing when $\|\boldsymbol{d}\|_2 > 0$, and treating the zero case directly, gives $\|\boldsymbol{d}\|_2^2 \le nQ_n$. Thus, we have

$$2A\widehat{A}(1-\rho_{\mathrm{S}}) = \|\boldsymbol{d}\|_2^2 - (A-\widehat{A})^2 \le nQ_n,$$

and therefor $1 - \rho_{\mathrm{S}} \le \Gamma_n \frac{Q_n}{M}$.

Consider $t > 0$. If $\xi_{ij} \neq 0$, the target and predicted signs differ, and thus $s_{ij}(\widehat{y}_i - \widehat{y}_j) \le 0$. This includes reversals and comparisons tied in exactly one of the two vectors. It implies $\ell_{ij} \ge \log 2$. Since $|\xi_{ij}| \le 1$, we have $\xi_{ij}^2 \le \ell_{ij}/\log 2$ for every comparison. Therefore, we have

$$\frac{1}{M}\sum_{\substack{i<j \\ |y_i - y_j| \le t}} \xi_{ij}^2 \le \min\left\{F_n(t), \frac{\mathcal{L}_{\mathrm{rank}}^{\le t}}{\log 2}\right\}.$$

For a comparison with $|y_i - y_j| > t$ and $\xi_{ij} \neq 0$, the predicted order is reversed or tied, and thus $|\varepsilon_i - \varepsilon_j| \ge |y_i - y_j| > t$. It then follows that

$$\frac{1}{M}\sum_{\substack{i<j \\ |y_i - y_j| > t}} \xi_{ij}^2 \le \frac{1}{Mt^2}\sum_{i<j}(\varepsilon_i - \varepsilon_j)^2 = \frac{n\sum_i \varepsilon_i^2 - (\sum_i \varepsilon_i)^2}{Mt^2} \le \frac{2n}{(n-1)t^2} r_n \le \frac{2n\,C_\beta(B)}{(n-1)t^2}\mathcal{L}_{\mathrm{reg}}.$$

Finally, without ties both rank vectors are permutations of $(1, \dots, n)$, and thus $A^2 = \widehat{A}^2 = \frac{n(n^2-1)}{12}$. Substitution into the definition of Γ_n then gives $\Gamma_n = 3n/(n+1)$. □

Theorem 13. *Fix one relation and a trained metanetwork. Draw a query and a candidate list of fixed size $K \ge 2$ from a specified joint distribution. All expectations and probabilities below are conditional on the trained metanetwork and are taken over this query-list distribution.*

For $j = 1, \dots, K$, let Y_j and $\widehat{Y}_j$ be the real-valued target and predicted relation values for candidate j. Denote $\varepsilon_j = \widehat{Y}_j - Y_j$ and $r_K = \mathbb{E}\left[\frac{1}{K}\sum_{j=1}^K \varepsilon_j^2\right] < \infty$. Let $\widehat{j} \in \arg\min_{1\le j\le K} \widehat{Y}_j$ and $j_\star \in \arg\min_{1\le j\le K} Y_j$, using fixed measurable tie-breaking rules, and define the retrieval regret as $\mathcal{R} = Y_{\widehat{j}} - Y_{j_\star}$. Then, we have $\mathbb{E}[\mathcal{R}^2] \le 2Kr_K$, and $0 \le \mathbb{E}[\mathcal{R}] \le \sqrt{2Kr_K}$.

If the best target candidate is unique almost surely, let

$$\Delta = Y_{(2)} - Y_{(1)} > 0 \qquad \textit{almost surely}, \tag{56}$$

where $Y_{(1)} \le Y_{(2)}$ are the smallest two target order statistics. Then, for every $\gamma > 0$, we have

$$\Pr\{\widehat{j} \neq j_\star\} \le \min\left\{1,\ \Pr\{\Delta \le \gamma\} + \frac{2Kr_K}{\gamma^2}\right\}. \tag{57}$$

If in addition $Y_j \in [0, B_{\mathrm{tar}}]$ almost surely for every j, then we have $\mathbb{E}[\mathcal{R}] \le \min\{B_{\mathrm{tar}}, \sqrt{2Kr_K}\}$.

Proof. Consider a fixed realized query and candidate list. By the optimality of $j_\star$ and $\widehat{j}$, we have

$$0 \le \mathcal{R} = Y_{\widehat{j}} - Y_{j_\star} = \widehat{Y}_{\widehat{j}} - \widehat{Y}_{j_\star} + \varepsilon_{j_\star} - \varepsilon_{\widehat{j}} \le \varepsilon_{j_\star} - \varepsilon_{\widehat{j}}.$$

If $\widehat{j} = j_\star$, then $\mathcal{R} = 0$. Otherwise the two indices are different, and thus

$$\mathcal{R}^2 \le (\varepsilon_{j_\star} - \varepsilon_{\widehat{j}})^2 \le 2(\varepsilon_{j_\star}^2 + \varepsilon_{\widehat{j}}^2) \le 2\sum_{j=1}^K \varepsilon_j^2,$$

where the last bound also holds when the indices are the same. Taking expectations and applying Cauchy-Schwarz, then we have $\mathbb{E}[\mathcal{R}^2] \le 2Kr_K$ and $\mathbb{E}[\mathcal{R}] \le \sqrt{\mathbb{E}[\mathcal{R}^2]} \le \sqrt{2Kr_K}$.

Now, suppose that the best target candidate is unique almost surely. On the event $\{\widehat{j} \neq j_\star\}$, the selected target is at least the second-smallest target, and thus we have $\mathcal{R} \ge \Delta$. Therefore, for every

$\gamma > 0$, we have $\{\widehat{j} \neq j_\star\} \subseteq \{\Delta \leq \gamma\} \cup \{\mathcal{R} > \gamma\}$. By the union bound and Markov's inequality, we have

$$\begin{aligned}\Pr\{\widehat{j} \neq j_\star\} &\leq \Pr\{\Delta \leq \gamma\} + \Pr\{\mathcal{R} > \gamma\} \\ &\leq \Pr\{\Delta \leq \gamma\} + \frac{\mathbb{E}[\mathcal{R}^2]}{\gamma^2} \\ &\leq \Pr\{\Delta \leq \gamma\} + \frac{2Kr_K}{\gamma^2}.\end{aligned}$$

□

We analyze a two-stage use of the metanetwork for stitching-candidate selection. First, predicted relation values determine a list of candidate pairs. Second, adapters are fitted for the retained candidates, and an independent validation sample is selected among the resulting stitched models. Model stitching assesses functional compatibility through the performance of a fitted composition (Bansal et al., 2021; Csiszárik et al., 2021). An activation-based relation is a screening surrogate and thus transferring a distance guarantee to stitching requires a separate surrogate-fidelity assumption.

Let $\mathcal{C}$ be a nonempty finite candidate bank. Fix a candidate-wise adapter-fitting procedure, its training data, and its random seeds. For each $c \in \mathcal{C}$, let f_c denote the stitched model produced by that procedure and define its population risk under the task law D by

$$R_c = \int \ell(f_c(x), y)\, d\mathsf{D}(x, y). \tag{58}$$

For candidates not fitted in deployment, f_c is the potential output of the same specified procedure. Thus, the reference oracle is over these procedure-specific fitted models, not over all possible adapters. Let $\mathcal{F}_0$ collect all pre-validation information, including these potential fitted models, the metanetwork outputs, and the list.

Theorem 14. *Assume $0 \leq \ell \leq B_\ell < \infty$. Let $\mathcal{S} \subseteq \mathcal{C}$ be a nonempty, $\mathcal{F}_0$-measurable list, and write $s = |\mathcal{S}|$. Let $\mathcal{V}_n = ((X_i, Y_i))_{i=1}^n \sim \mathsf{D}^n$ for $n \geq 1$ be independent of $\mathcal{F}_0$. Specifically, this validation sample is not used to construct the list or to fit or otherwise adapt the candidate models. Define*

$$\widehat{R}_c = \frac{1}{n}\sum_{i=1}^n \ell(f_c(X_i), Y_i), \qquad \widehat{c} \in \arg\min_{c \in \mathcal{S}} \widehat{R}_c, \tag{59}$$

using measurable tie-breaking, and let $R_\star = \min_{c\in\mathcal{C}} R_c$ and $B_{\text{screen}} = \min_{c\in\mathcal{S}} R_c - R_\star$. For $0 < \alpha < 1$, let $a_n(s, \alpha) = B_\ell\sqrt{\frac{\log(2s/\alpha)}{2n}}$. Then, conditionally on $\mathcal{F}_0$, with probability at least $1-\alpha$ over $\mathcal{V}_n$, we have

$$R_{\widehat{c}} - R_\star \leq B_{\text{screen}} + 2a_n(s, \alpha) \quad \text{and} \quad \mathbb{E}[R_{\widehat{c}} - R_\star \mid \mathcal{F}_0] \leq B_{\text{screen}} + B_\ell\sqrt{\frac{\log s}{2n}}. \tag{60}$$

Suppose additionally that δ_c and $\widehat{\delta}_c$ are, respectively, the true and predicted relation values used for screening, all measurable with respect to $\mathcal{F}_0$. Let $F : I \to \mathbb{R}$ be nondecreasing and L_F-Lipschitz on an interval I containing every δ_c, and assume that $\max_{c\in\mathcal{C}} |R_c - F(\delta_c)| \leq \zeta$ for some $\zeta \geq 0$. Define the relation-screening regret $A_\delta(\mathcal{S}) = \min_{c\in\mathcal{S}} \delta_c - \min_{c\in\mathcal{C}} \delta_c$. Then, deterministically conditional on $\mathcal{F}_0$, we have $B_{\text{screen}} \leq L_F A_\delta(\mathcal{S}) + 2\zeta$. Specifically, suppose that $\max_{c\in\mathcal{C}} |\widehat{\delta}_c - \delta_c| \leq E$ for some $E \geq 0$. If $\mathcal{S}$ contains at least one minimizer of $\widehat{\delta}_c$ over $\mathcal{C}$, then $B_{\text{screen}} \leq 2L_F E + 2\zeta$. If $\mathcal{S}$ contains a minimizer of δ_c, then instead $B_{\text{screen}} \leq 2\zeta$. The latter condition is guaranteed by

$$\mathcal{S} \supseteq \mathcal{S}_{\text{safe}}(E) := \left\{c \in \mathcal{C} : \widehat{\delta}_c \leq \min_{d\in\mathcal{C}} \widehat{\delta}_d + 2E\right\}. \tag{61}$$

Therefore, a list containing a predicted relation minimizer satisfies $R_{\widehat{c}} - R_\star \leq 2L_F E + 2\zeta + 2a_n(s, \alpha)$. Retaining the complete safe set instead gives $R_{\widehat{c}} - R_\star \leq 2\zeta + 2a_n(s, \alpha)$.

Proof. Condition on $\mathcal{F}_0$. The list and all fitted models are now fixed, and the validation observations remain independent with law D. If $B_\ell = 0$, the validation bounds are immediate and for their derivation below, assume $B_\ell > 0$.

For a random variable $Z \in [0, B_\ell]$, the centered log moment generating function has second derivative $\text{Var}_t(Z) \leq B_\ell^2/4$ under exponential tilting. Since its value and first derivative vanish at zero,

integration has $\mathbb{E}\exp(t(Z-\mathbb{E}Z)) \le \exp(t^2 B_\ell^2/8)$ for $t \in \mathbb{R}$. Independence across validation examples and exponential Markov inequality give

$$\Pr\{|\widehat{R}_c - R_c| > a \mid \mathcal{F}_0\} \le 2\exp(-2na^2/B_\ell^2), \qquad a > 0.$$

A union bound over the s retained candidates has $\max_{c\in\mathcal{S}} |\widehat{R}_c - R_c| \le a_n(s,\alpha)$ with conditional probability at least $1-\alpha$. For a fixed population minimizer $c_{\mathcal{S}} \in \arg\min_{c\in\mathcal{S}} R_c$, we have

$$R_{\widehat{c}} \le \widehat{R}_{\widehat{c}} + a_n(s,\alpha) \le \widehat{R}_{c_{\mathcal{S}}} + a_n(s,\alpha) \le R_{c_{\mathcal{S}}} + 2a_n(s,\alpha).$$

For the expectation bound, let $Z_{\max} = \max_{c\in\mathcal{S}}(R_c - \widehat{R}_c)$. By the empirical optimality, we have $R_{\widehat{c}} - R_{c_{\mathcal{S}}} \le Z_{\max} + \widehat{R}_{c_{\mathcal{S}}} - R_{c_{\mathcal{S}}}$. The difference $\widehat{R}_{c_{\mathcal{S}}} - R_{c_{\mathcal{S}}}$ has zero conditional mean. The same exponential-moment bound gives, for $t > 0$, we have

$$\mathbb{E}[Z_{\max} \mid \mathcal{F}_0] \le \frac{1}{t}\log\left(\sum_{c\in\mathcal{S}} \mathbb{E}[e^{t(R_c - \widehat{R}_c)} \mid \mathcal{F}_0]\right) \le \frac{\log s}{t} + \frac{tB_\ell^2}{8n}.$$

For $s > 1$, optimizing over t gives $B_\ell\sqrt{\log s/(2n)}$. For $s = 1$, selection adds no error and the same bound is zero.

Now, we choose $c_{\mathcal{S},\delta} \in \arg\min_{c\in\mathcal{S}} \delta_c$, $c_\delta \in \arg\min_{c\in\mathcal{C}} \delta_c$, and $c_R \in \arg\min_{c\in\mathcal{C}} R_c$, then we have

$$\begin{aligned}
\min_{c\in\mathcal{S}} R_c &\le R_{c_{\mathcal{S},\delta}} \\
&\le F(\delta_{c_{\mathcal{S},\delta}}) + \zeta \\
&\le F(\delta_{c_\delta}) + L_F A_\delta(\mathcal{S}) + \zeta \\
&\le F(\delta_{c_R}) + L_F A_\delta(\mathcal{S}) + \zeta \\
&\le R_\star + L_F A_\delta(\mathcal{S}) + 2\zeta.
\end{aligned}$$

If $\widetilde{c} \in \mathcal{S}$ minimizes the predicted relation value over $\mathcal{C}$, then

$$\delta_{\widetilde{c}} \le \widehat{\delta}_{\widetilde{c}} + E \le \widehat{\delta}_{c_\delta} + E \le \delta_{c_\delta} + 2E.$$

Thus $A_\delta(\mathcal{S}) \le 2E$. If $c_\delta \in \mathcal{S}$, then $A_\delta(\mathcal{S}) = 0$, giving $B_{\text{screen}} \le 2\zeta$.

For any true relation minimizer c_δ and any predicted minimizer $\widetilde{c}$, we have

$$\widehat{\delta}_{c_\delta} \le \delta_{c_\delta} + E \le \delta_{\widetilde{c}} + E \le \widehat{\delta}_{\widetilde{c}} + 2E.$$

Therefore every true relation minimizer lies in $\mathcal{S}_{\text{safe}}(E)$. □

D Experimental setup and details

D.1 Base-model populations and training

Datasets and model populations. We consider four dataset–architecture combinations: ResNets (He et al., 2016; Zagoruyko & Komodakis, 2016) and Vision Transformers (ViTs) (Dosovitskiy et al., 2021) trained on CIFAR-10, and the same two architecture families trained on CIFAR-100. For the CIFAR datasets, we use the standard data augmentation, consisting of random flips, random translations, and Cutout. For the ResNet family, we vary both network depth and width. Specifically, we consider ResNet20, ResNet26, ResNet44, and ResNet56, each with width multipliers of $\times 1$, $\times 2$, $\times 4$, and $\times 8$. Across these configurations, we collect a total of 20,000 independently trained ResNet models. We train each model for 300 epochs using stochastic gradient descent (SGD) with a momentum of 0.9, a batch size of 128, an initial learning rate of 0.1, and a weight decay of 5×10^{-4}. The learning rate is decayed according to a cosine schedule throughout training. For the ViT family, we vary the number of transformer blocks, attention heads, and per-head embedding dimensions. Specifically, we consider models with 2, 3, and 6 transformer blocks, 2, 3, and 6 attention heads, and per-head dimensions of 16, 32, and 64. Across these architectural configurations, we collect a total of 13,500 independently trained ViT models. We train each ViT for 300 epochs using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, a batch size of 128, and a weight decay of 0.05. The learning rate is linearly increased from 10^{-6} to 3×10^{-4} during the first 10 epochs and subsequently decayed to 10^{-6} using a cosine annealing schedule over the remaining epochs.

Model population split. We partition models into disjoint meta-training, validation, and test populations into 50%, 25%, and 25%, respectively, before constructing representation-site pairs. Site pairs are constructed only within their respective partitions.

D.2 Representation sites

A representation site consists of a block-output location and a fixed rule for extracting one feature vector per reference image. Sites are identified by their position in the forward computation.

ResNet sites. We select the output of every complete residual block in the three-stage CIFAR ResNets. For a model of depth d_m, each stage contains $(d_m - 2)/6$ two-convolution residual blocks. Thus, ResNet20, ResNet26, ResNet44, and ResNet56 contribute 9, 12, 21, and 27 sites, respectively. The width multiplier changes the channel dimensions but not the number of sites. Each site is identified by its stage index and block index within that stage. For the post-activation blocks specified here, we record the block output after the shortcut addition and final ReLU, rather than an intermediate convolution or normalization output. We average the resulting feature map over its spatial dimensions, separately for each channel, to obtain one feature vector per image. Its dimension equals the output channel count of the selected block. Spatial positions are neither concatenated nor treated as separate reference examples. The initial stem, final pooling operation, and classifier are not additional sites.

ViT sites. We select the output of every complete Transformer block. The configurations with 2, 3, and 6 blocks have 2, 3, and 6 sites, respectively, independently of the number of attention heads and the per-head attention dimension. For the pre-layer-normalized blocks specified here, the selected tensor is the output after the MLP residual addition. The final block is observed before the separate final encoder normalization and classification head. At each block, we extract the class-token vector at token index zero. Its feature dimension is the residual-stream token embedding dimension, not the per-head attention dimension. Patch tokens are not pooled, concatenated, or treated as separate examples. The patch-embedding output, individual attention heads, and classifier logits are not additional sites.

Reference matrices and preprocessing. We compute representation relations using a shared reference sample $\boldsymbol{X}_{\mathrm{ref}} = (\boldsymbol{x}_i)_{i=1}^{n}$ of $n = 1{,}000$ images. For each representation site u, we stack the extracted feature vectors into the activation matrix with one row per reference image and one column per feature coordinate. The reference images and their ordering are identical across all models and sites evaluated on the same reference sample. All activations are collected with the base models in evaluation mode. Site extraction applies only the spatial averaging specified for ResNets or the class-token selection specified for ViTs. Any relation-specific centering, normalization, neighborhood construction, or fitted transformation is applied subsequently as part of the corresponding target-construction procedure (App. B). The validation is determined separately for each relation because the conditions under which a relation is well defined differ across relation types. For a given h, all predictors are evaluated on the same set of valid site pairs. We report the number of excluded sites and site pairs for each relation. We do not remove reference examples independently for individual site pairs, since doing so would change the underlying reference sample across comparisons. In addition, we use batch aggregation as in Nguyen et al. (2021); Huh et al. (2024).

D.3 Method construction

Our method's central point is that all learned estimators are supervised using labeled pairs. The terms independent embedding and joint comparison refer to how a labeled pair is processed by the metanetwork. There are two levels of learning: *(i)* each base model is trained in the ordinary way on its task, and *(ii)* the base models are frozen and treated as *data* for a metanetwork. One meta-training instance is constructed by the following sequence.

1. Select two frozen base models and a representation site in each.
2. Extract the architecture and trained parameters of the computation upstream of each site.
3. Run the frozen base models on the data examples and compute the activation-defined target distances.
4. Present the two parameterized computations and the target distances to the metanetwork loss.

D.3.1 A TWO-LAYER EXAMPLE

Consider the two-layer linear network

$$\boldsymbol{h} = \boldsymbol{W}_1 \boldsymbol{x}, \qquad \boldsymbol{y} = \boldsymbol{W}_2 \boldsymbol{h}, \tag{62}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{h} \in \mathbb{R}^p$, $\boldsymbol{y} \in \mathbb{R}^q$, $\boldsymbol{W}_1 \in \mathbb{R}^{p \times d}$, and $\boldsymbol{W}_2 \in \mathbb{R}^{q \times p}$. The hidden activation $\boldsymbol{h}$ defines one representation site, say u. Its upstream computation $\mathcal{C}_u$ contains the input interface and the parameterized map $\boldsymbol{W}_1$. The output $\boldsymbol{y}$ defines a second site, say v, whose upstream computation $\mathcal{C}_v$ contains both linear maps and implements the composed transformation $\boldsymbol{W}_2 \boldsymbol{W}_1$. Now suppose that site u is compared with a representation site u' from another independently trained and frozen network. Evaluating the two sites on the same reference sample $\boldsymbol{X}$ produces activation matrices $\boldsymbol{H}_u(\boldsymbol{X})$ and $\boldsymbol{H}_{u'}(\boldsymbol{X})$. An activation-derived relation δ_{h}, such as CKA, then defines the target

$$y^{(\mathsf{h})}_{uu'} = \delta_{\mathsf{h}} \left(\boldsymbol{H}_u(\boldsymbol{X}), \boldsymbol{H}_{u'}(\boldsymbol{X}) \right). \tag{63}$$

The resulting meta-learning example is $(\mathcal{C}_u, \mathcal{C}_{u'}, y^{(\mathsf{h})}_{uu'})$. In this simplified setting, PARIS-Ind processes the two upstream computations independently to obtain reusable site embeddings,

$$\boldsymbol{z}_u = \mathrm{Enc}_\phi(\mathcal{C}_u), \qquad \boldsymbol{z}_{u'} = \mathrm{Enc}_\phi(\mathcal{C}_{u'}), \tag{64}$$

and predicts their relation from a relation-specific distance between these embeddings. By contrast, PARIS-Joint processes $\mathcal{C}_u$ and $\mathcal{C}_{u'}$ jointly, allowing information from the two parameterized computations to interact before the relation is predicted. Thus, this interaction can capture pair-dependent relationships between the two parameter matrices that cannot be represented by processing each site independently before comparison.

D.3.2 COMPUTATION GRAPH

For each representation site $u = (m, s)$, we have $\mathcal{C}_u = \mathcal{C}_m[\mathrm{Anc}_{\mathcal{C}_m}(u) \cup u]$. This graph includes trained parameters and state used during inference. We encode this computation into a graph

$$G_u = \Psi(\mathcal{C}_u) = (V_u, E_u, \boldsymbol{X}_u, \boldsymbol{A}_u, t_u), \tag{65}$$

where V_u and E_u are nodes and edges, $\boldsymbol{X}_u$ and $\boldsymbol{A}_u$ store their attributes, and t_u marks the target coordinates. Following Lim et al. (2024a), we attach each numerical parameter to the coordinates it connects instead of treating a whole parameter tensor as one feature vector. We also encode operator type, inference state, and the rule used to extract the target representation. The graph describes the trained model, and its activations are not included for graph features.

Graph structure. Coordinate nodes represent feature or channel dimensions at tensor boundaries. Operator nodes represent computations such as linear maps, convolutions, normalization, pooling, and nonlinearities. We add position, constant, attention-head, or normalization-group nodes only when the computation requires them. Edges describe both structure and numerical parameters. For example, a linear edge carries one weight, a convolution edge carries one kernel value and its offset, and bias edges connect a constant node to output coordinates. We also use structural edges to preserve input–output relations, residual connections, coordinate mappings, and normalization groups. We keep the inference state that affects the output. For example, BatchNorm includes its learned scale and bias, running mean and variance, and ϵ. We do not include training-only state that has no effect during evaluation.

Attention. For attention, we represent each head through its query-key and value-output circuits:

$$\boldsymbol{C}^{\mathrm{QK}}_a = \boldsymbol{W}_{Q,a} \boldsymbol{W}^\top_{K,a}, \qquad \boldsymbol{C}^{\mathrm{VO}}_a = \boldsymbol{W}_{V,a} \boldsymbol{W}_{O,a}. \tag{66}$$

We store these circuit values directly on graph edges and group the QK and VO circuits by attention head. This keeps their functional relation.

Target readout and message passing. We encode the rule used to extract the target representation as a final readout operation. This rule can select, for example, the full tensor, a class token, or an average over tokens or spatial positions. We mark only the output coordinates of this readout as targets. After R message-passing rounds, we combine the mean target state with the mean state of the full graph. We add a reverse message-passing edge for every forward edge. Reverse edges share the same numerical attributes but have a separate direction flag. They support information flow in both directions and do not represent inverse operations. Finally, we normalize numerical attributes using statistics from the meta-training graphs only. We keep all represented parameters, including zeros, and do not sample or prune nodes or edges.

D.3.3 ARCHITECTURE

We use the same fixed-width graph backbone for PARIS-Ind and PARIS-Joint. They share the site-graph schema and architecture, not parameter values. The backbone uses node width $d = 128$, message width 128, and six message-passing rounds. Each categorical field has an embedding width of 8. The node input MLP has dimensions $d_x \to 128 \to 128$, each round-specific edge-gate MLP has dimensions $d_e \to 128 \to 128$, and each node-update MLP has dimensions $384 \to 256 \to 128$. We use GELU nonlinearities, concatenate the sum and mean of incoming messages, and apply LayerNorm after each residual node update. Different message-passing rounds use different parameters. We use no dropout and do not sample nodes, edges, or tokens.

For PARIS-Ind, the readout has dimensions $256 \to 256 \to 128$ and produces a persistent site embedding $\boldsymbol{z}_u \in \mathbb{R}^{128}$. Each comparison rule $h \in \mathcal{H}$ uses a bias-free metric head $\boldsymbol{W}_h \in \mathbb{R}^{64\times 128}$ followed by the monotone calibration $1 - \exp(-\varepsilon_h d)$.

For PARIS-Joint, the graph backbone uses width 128. The comparator keeps operator tokens, attention-head tokens, and two summary tokens per site. It uses two bidirectional cross-attention blocks with four heads each, model width 128, and per-head width 32. Each cross-attention feed-forward network has dimensions $128 \to 512 \to 128$. The contextual readout has dimensions $256 \to 256 \to 128$, and the final symmetric prediction head has dimensions $384 \to 256 \to H$. The joint estimator uses a sigmoid output rather than the metric heads and exponential calibrations of the independent estimator. These widths do not depend on the depth, width, or number of attention heads of the represented base model, so the same metanetwork can process the variable-size ResNet and ViT site graphs considered in our experiments.

Input attributes. Each categorical node or edge field has its own learned lookup table of width 8. If a field does not apply, it contributes a fixed zero vector. Let F_x and F_e denote the numbers of categorical node and edge fields, and let n_x and n_e count the normalized numerical values, flags, and presence masks. The resulting node and edge widths are

$$d_x = n_x + 8F_x, \qquad d_e = n_e + 8F_e. \tag{67}$$

We fix the field vocabularies, their ordering, and all normalization constants from the feature manifest. We do not fit vocabularies or normalization statistics on validation or test graphs. The manifest determines the actual values of d_x, d_e, and the vocabulary sizes.

For a node feature vector $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$, we initialize

$$\boldsymbol{h}_i^{(0)} = \mathrm{LN}_{\mathrm{in}}\left(\boldsymbol{A}_{\mathrm{in},2}\ \mathrm{GELU}\left(\boldsymbol{A}_{\mathrm{in},1}\boldsymbol{x}_i + \boldsymbol{b}_{\mathrm{in},1}\right) + \boldsymbol{b}_{\mathrm{in},2}\right), \tag{68}$$

where $\boldsymbol{A}_{\mathrm{in},1} \in \mathbb{R}^{d\times d_x}$ and $\boldsymbol{A}_{\mathrm{in},2} \in \mathbb{R}^{d\times d}$. All node types use the same MLP, and their types enter through the input attributes.

Message-passing rounds. We use $R = 6$. For a directed edge e with source $s(e)$, target $t(e)$, and attributes $\boldsymbol{a}_e \in \mathbb{R}^{d_e}$, round r computes

$$\boldsymbol{g}_e^{(r)} = \boldsymbol{B}_{e,2}^{(r)}\,\mathrm{GELU}\left(\boldsymbol{B}_{e,1}^{(r)}\boldsymbol{a}_e + \boldsymbol{b}_{e,1}^{(r)}\right) + \boldsymbol{b}_{e,2}^{(r)}, \tag{69}$$

$$\boldsymbol{m}_e^{(r)} = \left(W^{(r)}\boldsymbol{h}_{s(e)}^{(r)}\right) \odot \boldsymbol{g}_e^{(r)}. \tag{70}$$

Here, $\boldsymbol{B}_{e,1}^{(r)} \in \mathbb{R}^{d\times d_e}$, $\boldsymbol{B}_{e,2}^{(r)} \in \mathbb{R}^{d\times d}$, and $\boldsymbol{W}^{(r)} \in \mathbb{R}^{d\times d}$. The source projection $\boldsymbol{W}^{(r)}$ has no bias. The edge MLP has biases in both affine layers, no normalization, and no final sigmoid. Its output is a signed gate. Edge attributes remain fixed across rounds, and we do not maintain a separate recurrent edge state. For node j, let $\mathcal{E}^-(j)$ denote the multiset of incoming edges, including reverse and parallel edges. We compute both their sum and mean:

$$\boldsymbol{s}_j^{(r)} = \sum_{e\in\mathcal{E}^-(j)} \boldsymbol{m}_e^{(r)}, \qquad \bar{\boldsymbol{m}}_j^{(r)} = \begin{cases} \boldsymbol{s}_j^{(r)}/|\mathcal{E}^-(j)|, & |\mathcal{E}^-(j)| > 0, \\ \boldsymbol{0}, & |\mathcal{E}^-(j)| = 0. \end{cases} \tag{71}$$

We concatenate the current node state with both aggregates and update

$$\Delta\boldsymbol{h}_j^{(r)} = B_{n,2}^{(r)}\,\mathrm{GELU}\left(\boldsymbol{B}_{n,1}^{(r)}\left[\boldsymbol{h}_j^{(r)} \| \boldsymbol{s}_j^{(r)} \| \bar{\boldsymbol{m}}_j^{(r)}\right] + \boldsymbol{b}_{n,1}^{(r)}\right) + \boldsymbol{b}_{n,2}^{(r)}, \tag{72}$$

$$\boldsymbol{h}_j^{(r+1)} = \mathrm{LN}_r\left(\boldsymbol{h}_j^{(r)} + \Delta\boldsymbol{h}_j^{(r)}\right), \tag{73}$$

where $\boldsymbol{B}_{n,1}^{(r)} \in \mathbb{R}^{2d\times 3d}$ and $\boldsymbol{B}_{n,2}^{(r)} \in \mathbb{R}^{d\times 2d}$. Each round has its own source projection, edge MLP, node MLP, and LayerNorm. Within a round, these parameters are shared across all graph locations and all site graphs. The categorical lookup tables and the input encoder are shared across rounds.

LayerNorm acts on the d features of each node independently:

$$\mathrm{LN}(\boldsymbol{a}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{a} - \mu(\boldsymbol{a})\mathbf{1}}{\sqrt{d^{-1}\sum_{j=1}^{d}(a_j - \mu(\boldsymbol{a}))^2 + 10^{-5}}} + \boldsymbol{\beta}, \qquad \mu(\boldsymbol{a}) = d^{-1}\sum_{j=1}^{d} a_j. \tag{74}$$

Pooling and site embedding. Let $S_u \subseteq V_u$ be the nonempty set of coordinate nodes marked by the target readout. After the final message-passing round, we compute

$$\boldsymbol{p}_u = \frac{1}{|S_u|}\sum_{i\in S_u} \boldsymbol{h}_i^{(R)}, \qquad \boldsymbol{g}_u = \frac{1}{|V_u|}\sum_{i\in V_u} \boldsymbol{h}_i^{(R)}. \tag{75}$$

The whole-graph mean includes every node type and excludes padding. The target mean uses only the marked target-coordinate nodes.

The independent encoder maps the two pooled vectors to

$$\boldsymbol{z}_u = \boldsymbol{A}_{\rho,2}\,\mathrm{GELU}\left(\boldsymbol{A}_{\rho,1}[\boldsymbol{p}_u \| \boldsymbol{g}_u] + b_{\rho,1}\right) + b_{\rho,2} \in \mathbb{R}^k, \tag{76}$$

with dimensions $2d \to 2d \to k$, i.e., $256 \to 256 \to 128$.

For each comparison rule $h \in \mathcal{H}$, a bias-free matrix $\boldsymbol{W}_h \in \mathbb{R}^{r_h\times k}$ with $r_h = 64$ defines

$$d_{\phi,h}(u,v) = \left\|\boldsymbol{W}_h(\boldsymbol{z}_u - \boldsymbol{z}_v)\right\|_2. \tag{77}$$

We optimize $\boldsymbol{W}_h$ directly and do not constrain its positivity, orthogonality, spectral norm, or Frobenius norm. The induced matrix $\boldsymbol{M}_h = \boldsymbol{W}_h^\top \boldsymbol{W}_h$ is positive semidefinite by construction and has rank at most 64. Thus, $d_{\phi,h}$ is a pseudometric: it is symmetric, satisfies the triangle inequality, and gives $d_{\phi,h}(u,u) = 0$, but two distinct sites may still have zero distance. We use the ordinary Euclidean norm and a zero subgradient at the zero vector.

We map the latent distance to the target dissimilarity scale using

$$\widehat{y}_{uv}^{(h)} = 1 - \exp\left[-\varepsilon_h d_{\phi,h}(u,v)\right], \tag{78}$$

$$\varepsilon_h = \mathrm{softplus}(\theta_h) + \varepsilon_{\min}, \qquad \varepsilon_{\min} = 10^{-6}, \tag{79}$$

where $\mathrm{softplus}(t) = \log(1 + e^t)$. Each comparison rule has one learned scalar θ_h. We initialize $\varepsilon_h = 1$. We evaluate it numerically as $-\operatorname{expm1}(-\varepsilon_h d_{\phi,h})$ with a stable softplus.

Let $\mathcal{B}$ be one batch of supervised site pairs and define the set of unique sites

$$\mathcal{U}_{\mathcal{B}} = \mathrm{unique}\left(\{u : (u,v) \in \mathcal{B}\} \cup \{v : (u,v) \in \mathcal{B}\}\right), \qquad M = |\mathcal{U}_{\mathcal{B}}|. \tag{80}$$

For each dataset-architecture setting, we train PARIS-Ind and PARIS-Joint on the same sequence of sampled batches. Each batch $\mathcal{B}$ contains $B = 64$ unordered pairs of distinct sites, and all sites come from the meta-training partition. For $M \geq 2$, let

$$\mu_j = \frac{1}{M}\sum_{u\in\mathcal{U}_{\mathcal{B}}} z_{u,j}, \qquad \sigma_j = \sqrt{\frac{1}{M}\sum_{u\in\mathcal{U}_{\mathcal{B}}}(z_{u,j} - \mu_j)^2}. \tag{81}$$

We use

$$\mathcal{L}_{\mathrm{var}} = \max\left(0, \eta - \frac{1}{k}\sum_{j=1}^{k}\sigma_j\right). \tag{82}$$

The standard deviation uses denominator M, not $M - 1$. When $M < 2$, we set $\mathcal{L}_{\mathrm{var}} = 0$. We compute the statistic from the unnormalized embeddings $\boldsymbol{z}_u$. It discourages global collapse.

The joint estimator PARIS-Joint first applies the same $d = 128$, $R = 6$ message-passing architecture to G_u and G_v separately. To keep the pairwise comparator tractable, we do not expose every coordinate node as a cross-attention token. Let $\mathcal{O}_u$ and $\mathcal{A}_u$ denote the operator and attention-head nodes of G_u. Before adding role embeddings, we use

$$\mathcal{T}_u^{\mathrm{raw}} = \{\boldsymbol{p}_u\}_{\mathrm{target}} \uplus \{\boldsymbol{g}_u\}_{\mathrm{global}} \uplus \{\boldsymbol{h}_i^{(R)} : i \in \mathcal{O}_u\} \uplus \{\boldsymbol{h}_i^{(R)} : i \in \mathcal{A}_u\}. \tag{83}$$

Each token receives a learned width-128 role embedding that marks one of four roles: target summary, global summary, operator, or attention head. The resulting number of tokens is

$$n_u = |\mathcal{O}_u| + |\mathcal{A}_u| + 2, \qquad T_u^{(0)} \in \mathbb{R}^{n_u \times 128}. \tag{84}$$

For ResNet sites, $|\mathcal{A}_u| = 0$. For a ViT site after block ℓ_u,

$$|\mathcal{A}_u| = \sum_{\ell=1}^{\ell_u} A_{\ell,\mathrm{base}}. \tag{85}$$

For a uniform-head ViT this becomes $\ell_u A_{\mathrm{base}}$. In the stated ViT population, the largest complete prefix contains at most $6 \times 6 = 36$ attention-head tokens.

Bidirectional cross-attention. We use $L_\times = 2$ cross-attention blocks. Each block has $a_\times = 4$ comparator heads, model width 128, and per-head query/key/value width 32.

For block ℓ, we first normalize the two token streams,

$$\boldsymbol{N}_u^{(\ell)} = \mathrm{LN}_{A,\ell}(\boldsymbol{T}_u^{(\ell)}), \qquad \boldsymbol{N}_v^{(\ell)} = \mathrm{LN}_{A,\ell}(\boldsymbol{T}_v^{(\ell)}). \tag{86}$$

For comparator head b, the update from v to u is

$$\mathrm{Att}_{b,\ell}(\boldsymbol{N}_u, \boldsymbol{N}_v) = \mathrm{softmax}\left(\frac{Q_{b,\ell}(\boldsymbol{N}_u)K_{b,\ell}(\boldsymbol{N}_v)^\top}{\sqrt{32}} + \boldsymbol{M}_{u\leftarrow v}\right) V_{b,\ell}(\boldsymbol{N}_v), \tag{87}$$

where the query, key, and value projections are affine and $\boldsymbol{M}_{u\leftarrow v}$ masks padded partner tokens. The two directions are updated by

$$\bar{\boldsymbol{T}}_u^{(\ell)} = \boldsymbol{T}_u^{(\ell)} + \mathrm{MHA}_\ell\left(\boldsymbol{N}_u^{(\ell)}, \boldsymbol{N}_v^{(\ell)}\right), \tag{88}$$

$$\bar{\boldsymbol{T}}_v^{(\ell)} = \boldsymbol{T}_v^{(\ell)} + \mathrm{MHA}_\ell\left(\boldsymbol{N}_v^{(\ell)}, \boldsymbol{N}_u^{(\ell)}\right), \tag{89}$$

$$\boldsymbol{T}_u^{(\ell+1)} = \bar{\boldsymbol{T}}_u^{(\ell)} + \mathrm{FFN}_\ell\left(\mathrm{LN}_{F,\ell}(\bar{\boldsymbol{T}}_u^{(\ell)})\right), \tag{90}$$

$$\boldsymbol{T}_v^{(\ell+1)} = \bar{\boldsymbol{T}}_v^{(\ell)} + \mathrm{FFN}_\ell\left(\mathrm{LN}_{F,\ell}(\bar{\boldsymbol{T}}_v^{(\ell)})\right). \tag{91}$$

The feed-forward network has dimensions $128 \to 512 \to 128$, uses GELU after the first affine layer, and has biases in both layers. We use no token self-attention inside the comparator. Each sublayer uses pre-LayerNorm with learned gain and bias and $\epsilon = 10^{-5}$. After the second cross-attention block, we apply one final shared LayerNorm.

Let $F_{u|v} = \mathrm{LN}_{\mathrm{out}}\left(\boldsymbol{T}_u^{(L_\times)}\right)$. We concatenate the final target-summary token with the mean of all valid tokens and compute

$$\boldsymbol{c}_{u|v} = \rho_J\left(F_{u|v,\mathrm{target}} \| \frac{1}{n_u}\sum_{t=1}^{n_u} F_{u|v,t}\right) \in \mathbb{R}^{128}. \tag{92}$$

The readout ρ_J has dimensions $256 \to 256 \to 128$ with one intermediate GELU. It is shared between the two sides. Let $\boldsymbol{c} = \boldsymbol{c}_{u|v}$ and $\boldsymbol{c}' = \boldsymbol{c}_{v|u}$. We form the symmetric pair feature

$$\boldsymbol{s}_{uv} = [\boldsymbol{c} + \boldsymbol{c}' \parallel |\boldsymbol{c} - \boldsymbol{c}'| \parallel \boldsymbol{c} \odot \boldsymbol{c}'] \in \mathbb{R}^{384}. \tag{93}$$

The final predictor is

$$\widehat{\boldsymbol{y}}_{uv} = \mathrm{sigmoid}\left(\boldsymbol{A}_{J,2}\,\mathrm{GELU}\left(\boldsymbol{A}_{J,1}\boldsymbol{s}_{uv} + \boldsymbol{b}_{J,1}\right) + \boldsymbol{b}_{J,2}\right) \in (0,1)^H, \tag{94}$$

where $\boldsymbol{A}_{J,1} \in \mathbb{R}^{256\times 384}$ and $\boldsymbol{A}_{J,2} \in \mathbb{R}^{H\times 256}$. The h-th coordinate is the predicted score $\widehat{y}_{uv}^{(h)}$ for comparison rule h.

Initialization. Except for the metric heads and calibration scalars described above, we initialize affine weight matrices with Xavier-uniform initialization with gain one (Glorot & Bengio, 2010) and set affine biases to zero. We initialize categorical embeddings and joint token-role embeddings independently from $\mathcal{N}(0, 0.02^2)$. LayerNorm gains start at one and biases at zero.

Objectives and optimizer. The objectives are

$$\mathcal{L}_{\mathrm{Ind}} = \frac{1}{H}\sum_{h=1}^{H}\left(\mathcal{L}_{\mathrm{reg}}^{(h)} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}}^{(h)}\right) + \lambda_{\mathrm{var}}\mathcal{L}_{\mathrm{var}}, \tag{95}$$

$$\mathcal{L}_{\mathrm{Joint}} = \frac{1}{H}\sum_{h=1}^{H}\left(\mathcal{L}_{\mathrm{reg}}^{(h)} + \lambda_{\mathrm{rank}}\mathcal{L}_{\mathrm{rank}}^{(h)}\right). \tag{96}$$

We use $\tau = 8$ and $\eta = 0.04$ and $\lambda_{\mathrm{rank}} = 0.14$ and $\lambda_{\mathrm{var}} = 0.5$. Both estimators use AdamW with weight decay coefficient 2×10^{-4}. The initial learning rate is 2×10^{-3} for PARIS-Ind and 1.5×10^{-3} for PARIS-Joint. After accumulating all gradients for one batch, we clip their global Euclidean norm to 5 and perform one optimizer step.

Trainable parameter counts. Let

$$P_{\mathrm{cat}} = 8\left(\sum_{f=1}^{F_x} K_{x,f} + \sum_{f=1}^{F_e} K_{e,f}\right) \tag{97}$$

count the learned categorical lookup entries, excluding fixed zero rows for inapplicable fields. With the biases and affine LayerNorm parameters above, the shared graph backbone contains $P_B = P_{\mathrm{cat}} + d\,d_x + d^2 + 4d + R\left(10d^2 + d\,d_e + 7d\right)$. The independent readout contains $P_\rho = 4d^2 + 2dk + 2d + k$, so $P_{\mathrm{Ind}} = P_B + P_\rho + \sum_{h=1}^{H}(r_h k + 1)$. The additional scalar for each comparison rule is its learned calibration parameter.

For PARIS-Joint, the four token-role embeddings add $4d$ parameters. Each cross-attention block adds $12d^2 + 13d$ parameters from the four attention projections, the $d \to 4d \to d$ feed-forward network, and two affine LayerNorms. The final token LayerNorm adds $2d$, the contextual readout adds P_ρ, and the symmetric prediction head adds $6dk + 2d + (2d+1)H$. Thus,

$$P_{\mathrm{Joint}}P_B + 4d + L_\times(12d^2 + 13d) + 2d + P_\rho + 6dk + 2d + (2d+1)H. \tag{98}$$

Edge-block processing. We use float32 for metanetwork parameters, numerical graph attributes, node states, and reductions. Attention-circuit values and static graph topology may remain in host memory and be transferred in directed-edge blocks of at most $C_E = 65{,}536$. For each block, we compute the categorical embeddings, edge gates, and messages and accumulate them into the destination-node sums. We compute the incoming degree over the complete graph, so the mean is formed only after all edge blocks have contributed. Edge blocking changes only the execution order.

We do not need to retain a complete $|E_u| \times d$ array of gates or messages. During training, we checkpoint message-passing rounds and recompute edge-block intermediates during the backward pass. Gradients from each block accumulate into the shared parameters and source-node states. Static graph attributes and categorical indices may be cached, but learned edge gates are recomputed after every parameter update.

D.3.4 Inference-time computational cost

We compare inference costs after the metanetwork has been trained and frozen. We compare two ways to obtain representation relations: PARIS predicts a relation from trained computations, whereas direct evaluation measures the relation from newly computed representations. We also include every computation needed to score a candidate model that has not been processed before. For PARIS, this includes graph construction and encoding. For direct evaluation, it includes feature extraction and processing required by the chosen relation. If a direct relation fits a readout to a candidate model's features, we also include that cost.

Fix one reference protocol and one relation h. Let $\mathcal{M}$ be the set of distinct models, $\mathcal{U}$ the set of N requested sites, and $\mathcal{P}$ the set of K distinct site pairs to score. Let n be the total number of distinct shared inputs required by the protocol. This count includes inputs used to fit or tune a readout when needed. Let $\boldsymbol{H}_u \in \mathbb{R}^{n\times p_u}$ denote the features extracted at site u. We count computation at fixed precision and assume conventional dense linear algebra.

Direct evaluation from representations. Let F_m denote the work per input required to run model m far enough to collect *all requested sites*, including the work required by their extraction rules. In

a sequential model, one traversal to the deepest requested site is sufficient. The model does not need a separate traversal for each site. For fixed input shapes, collecting the representations costs

$$\mathcal{F}(n) = n \sum_{m \in \mathcal{M}} F_m. \tag{99}$$

We pay this cost once for each model and reference sample, not once for each site pair. For an implementation that preprocesses each site, let $a_h(u)$ denote the work needed to prepare the comparison descriptor for site u after feature extraction. Let $b_h(u, v)$ denote the work needed to score one pair from the prepared quantities. Then

$$C_{\mathrm{dir}}^{(h)} = \mathcal{F}(n) + \sum_{u \in \mathcal{U}} a_h(u) + \sum_{(u,v) \in \mathcal{P}} b_h(u, v). \tag{100}$$

The preparation term includes any required centering, kernel construction, neighbor search, or readout fitting and evaluation. For a readout-based relation, feature extraction must also include the examples used to fit and tune the readout, not only the examples used to measure disagreement.

For the independent estimator, write

$$\boldsymbol{z}_u = \mathrm{Enc}_\phi(G_u) \in \mathbb{R}^k, \qquad \widetilde{\boldsymbol{z}}_{u,h} = \boldsymbol{W}_h \boldsymbol{z}_u \in \mathbb{R}^{k_h}. \tag{101}$$

Because the projection is linear, $\|\boldsymbol{W}_h(\boldsymbol{z}_u - \boldsymbol{z}_v)\|_2 = \|\widetilde{\boldsymbol{z}}_{u,h} - \widetilde{\boldsymbol{z}}_{v,h}\|_2$. We can project each site once and reuse the result across pairs. Let $\Gamma_{\mathcal{U}}$ denote the total work needed to build the required graphs and compute all $\boldsymbol{z}_u$, after reusing any common preprocessing that the implementation supports. A dense projection costs $O(kk_h)$ per site. The distance calculation and scalar calibration cost $O(k_h)$ per pair. Hence,

$$C_{\mathrm{Ind}}^{(h)} = \Gamma_{\mathcal{U}} + O\left(Nkk_h + Kk_h\right). \tag{102}$$

For one fixed relation, storing only the projected embeddings requires $O(Nk_h)$ values. If we evaluate several relations, all of them share the graph encoding, while their projection and scoring costs add across heads. Direct methods can share feature extraction across relations.

Let $v_u = |V_u|$ and $e_u = |E_u|$, including reverse edges. Consider the stated gated message-passing update with R rounds, hidden dimension d, and fixed-depth node-update MLPs with hidden widths of $O(d)$. At each round, computing Wh_i once for every node costs $O(v_u d^2)$. Computing edge gates and aggregating messages costs $O(e_u d)$. Let c_{edge} denote the total per-edge work of the distinct edge-MLP evaluations used across all rounds. Encoding one site then costs

$$\gamma_u = C_{\mathrm{aux},u} + O\left(e_u c_{\mathrm{edge}} + R(v_u d^2 + e_u d)\right), \tag{103}$$

where $C_{\mathrm{aux},u}$ covers graph construction, attribute preprocessing, node initialization, pooling, and the embedding readout. If the model uses the same static edge gates across rounds, we can cache them. If it uses different edge maps in different rounds, we must count each map separately. Note that a dense width-d hidden layer in the edge MLP costs $O(d^2)$ per edge for each distinct map.

Comparison example. Let $\bar{\boldsymbol{H}}_u$ denote the features after centering them over the n inputs, and assume that the centered Gram matrices are nonzero. In feature space,

$$s_{\mathrm{CKA}}(u, v) = \frac{\|\bar{\boldsymbol{H}}_u^\top \bar{\boldsymbol{H}}_v\|_F^2}{\|\bar{\boldsymbol{H}}_u^\top \bar{\boldsymbol{H}}_u\|_F \|\bar{\boldsymbol{H}}_v^\top \bar{\boldsymbol{H}}_v\|_F}. \tag{104}$$

Computing each denominator costs $O(np_u^2)$. For each pair, computing the cross-product costs $O(np_u p_v)$. If the features are cached, the total cost is

$$C_{\mathrm{CKA,feat}} = \mathcal{F}(n) + O\left(n \sum_{u \in \mathcal{U}} p_u^2 + n \sum_{(u,v) \in \mathcal{P}} p_u p_v\right). \tag{105}$$

We can instead prepare the normalized Gram descriptor

$$\boldsymbol{B}_u = \frac{\bar{\boldsymbol{H}}_u \bar{\boldsymbol{H}}_u^\top}{\|\bar{\boldsymbol{H}}_u \bar{\boldsymbol{H}}_u^\top\|_F}, \qquad s_{\mathrm{CKA}}(u, v) = \langle \boldsymbol{B}_u, \boldsymbol{B}_v \rangle_F. \tag{106}$$

Constructing this descriptor costs $O(n^2 p_u)$ per site, and comparing two descriptors costs $O(n^2)$ per pair. Thus,

$$C_{\mathrm{CKA,Gram}} = \mathcal{F}(n) + O\left(n^2 \sum_{u \in \mathcal{U}} p_u + Kn^2\right). \tag{107}$$

We center the features with their means. Converting the similarity to chordal dissimilarity adds only constant work per pair. Both forms compute the same full-sample relation. Which form is cheaper depends on n, the feature dimensions, the number of requested pairs, and the available memory.

For one relation, we assume $p_u = p$ for simplicity. We also assume that no candidate descriptors are cached before preparation, so each method pays its preparation cost once. For linear CKA with feature caching, preparation costs $\mathcal{F}(n) + O(Nnp^2)$, each pairwise comparison costs $O(np^2)$, and the cache requires $O(Nnp)$ space. For linear CKA with Gram caching, preparation costs $\mathcal{F}(n) + O(Nn^2p)$, each pairwise comparison costs $O(n^2)$, and the cache requires $O(Nn^2)$ space. For PARIS-Ind, preparation costs $\Gamma_{\mathcal{U}} + O(Nkk_h)$, each pairwise comparison costs $O(k_h)$, and the cache requires $O(Nk_h)$ space. They exclude model weights, encoder workspace, and the optional $O(K)$ space used to store output scores.

These costs apply to the stated full-sample implementations. Consider a CKA estimator that uses one shared partition of the data into batches of size at most b. If it builds and aggregates kernels only within each batch, kernel construction costs $O(nbp_u)$ per site and comparison costs $O(nb)$ per pair. These bounds exclude any extra work for bandwidth selection or repeated partitions. The cost of direct comparison depends on the relation. Cached sample-space GULP operators or normalized RBF kernels allow $O(n^2)$ comparison. MutualNN neighbor sets stored in example-index order allow $O(nk_{\mathrm{NN}})$ comparison. Cached hard readout predictions allow $O(n_{\mathrm{eval}})$ disagreement evaluation. We still include the cost of constructing these descriptors in $a_h(u)$.

The joint estimator cannot reuse a final site embedding independently of its comparison partner. Let Γ_{Joint} denote the reusable work for graph and token preparation, and let $j_h(u, v)$ denote the remaining work for pair (u, v). Then

$$C^{(h)}_{\mathrm{Joint}} = \Gamma_{\mathrm{Joint}} + \sum_{(u,v)\in\mathcal{P}} j_h(u, v). \tag{108}$$

Suppose sites u and v contain τ_u and τ_v tokens. For B dense bidirectional cross-attention blocks with total hidden width d_{J} and feed-forward widths of $O(d_{\mathrm{J}})$, the cross-attention blocks cost

$$O\left(B\left[(\tau_u + \tau_v)d_{\mathrm{J}}^2 + \tau_u\tau_v d_{\mathrm{J}}\right]\right) \tag{109}$$

per pair, in addition to the output head and any other work that depends on the pair. Therefore, the $O(k_h)$ cached-scoring bound for PARIS-Ind does not apply to PARIS-Joint. Avoiding evaluation on reference inputs is not, by itself, enough to show that PARIS-Joint scores pairs more cheaply than a cached direct method.

Let D_0 and I_0 denote the realized preparation costs of direct evaluation and PARIS-Ind. Let $\bar{b}_{\mathrm{dir}}$ and $\bar{b}_{\mathrm{Ind}}$ denote their average per-pair scoring costs. PARIS-Ind has lower total inference cost when $K\left(\bar{b}_{\mathrm{dir}} - \bar{b}_{\mathrm{Ind}}\right) > I_0 - D_0$. If $\bar{b}_{\mathrm{dir}} > \bar{b}_{\mathrm{Ind}}$, this condition is equivalent to $K > (I_0 - D_0)/(\bar{b}_{\mathrm{dir}} - \bar{b}_{\mathrm{Ind}})$.

If candidate descriptors are already cached, we exclude their preparation cost from the incremental comparison for *both* methods. A new query model still requires its own preparation and query–bank scoring, but we do not process the cached candidate models again. Exhaustive all-pairs scoring still requests $K = \Theta(N^2)$ scores under both methods. PARIS-Ind changes the cost of preparing each site and scoring each pair.

After training, PARIS does not need to evaluate a new candidate model on the reference inputs. PARIS-Ind then compares sites by computing distances in $\mathbb{R}^{k_h}$. Its deployment cost has no explicit dependence on the number of reference examples used to define the learned target. PARIS has an inference-time advantage only when graph preparation, encoding, and scoring cost less in total than the corresponding direct pipeline.

D.3.5 Relation to contrastive learning and supervised metric learning

Classical contrastive metric learning uses a binary label that marks two examples as similar or dissimilar and then attracts or repels their embeddings, often with a margin for dissimilar pairs (Hadsell et al., 2006). Modern contrastive objectives such as InfoNCE and SimCLR define positive views for an anchor and distinguish them from sampled negatives through a normalized log-softmax loss (Oord et al., 2018; Chen et al., 2020). Supervised contrastive learning replaces augmentation-based positives by examples with the same class label, but it still separates positives from negatives (Khosla et al., 2020).

Our supervision is different. For each comparison rule $h \in \mathcal{H}$ and site pair (u, v), we compute a continuous target dissimilarity

$$y_{uv}^{(h)} = \Delta_h(\boldsymbol{H}_u, \boldsymbol{H}_v) \in [0, 1]. \tag{110}$$

For example, a pair with score 0.08 should be closer than one with score 0.35, which should in turn be closer than one with score 0.82. Turning these values into binary positive and negative labels would discard both their scale and much of their ordering.

We train directly on the continuous scores. The regression term encourages the predicted value $\widehat{y}_{uv}^{(h)}$ to match $y_{uv}^{(h)}$, while the ranking term encourages the model to preserve the ordering between two supervised site pairs. The ranking loss does not define one pair as positive and the remaining pairs as negatives. Thus, PARIS-Ind is more naturally viewed as supervised metric learning with continuous target distances, or as distance distillation. PARIS-Joint is even less similar to standard contrastive learning: it predicts the score for a pair directly and does not need to embed every site into one shared metric space.

These differences also clarify the role of the output geometry. Binary contrastive learning, InfoNCE, and supervised contrastive learning primarily specify which examples should be close or far apart. In contrast, our objective uses the full target dissimilarity $y_{uv}^{(h)}$. For PARIS-Ind, this supervision is realized through the pseudometric $d_{\phi,h}(u, v)$ and its monotone calibration. For PARIS-Joint, the model predicts $\widehat{y}_{uv}^{(h)}$ with a pair-dependent comparator, without imposing a common metric geometry.

A contrastive variant of our method could potentially be defined. For a query site u and a candidate set $\mathcal{V}_u$, the target scores can define a soft neighborhood distribution

$$p_h(v \mid u) = \frac{\exp\left[-y_{uv}^{(h)}/T_y\right]}{\sum_{w \in \mathcal{V}_u} \exp\left[-y_{uw}^{(h)}/T_y\right]}. \tag{111}$$

For PARIS-Ind, the learned distances can define

$$q_{\phi,h}(v \mid u) = \frac{\exp[-d_{\phi,h}(u, v)/T_z]}{\sum_{w \in \mathcal{V}_u} \exp[-d_{\phi,h}(u, w)/T_z]}. \tag{112}$$

One could then minimize

$$\mathcal{L}_{\text{softcon}}^{(h)} = \mathbb{E}_u \, \mathrm{KL}(p_h(\cdot \mid u) \,\|\, q_{\phi,h}(\cdot \mid u)). \tag{113}$$

This objective would use normalized neighborhoods in a contrastive manner while preserving graded similarity through soft weights. It could serve as an auxiliary retrieval objective, and we leave it for future exploration. Unlike direct regression to $y_{uv}^{(h)}$, this objective mainly fixes relative neighborhood structure and does not directly recover the absolute target scale.

Verified reparameterizations provide another possible source of positive pairs. If a transformation γ leaves the computation unchanged, then G_u and γG_u describe equivalent computations. For the hidden-coordinate, feed-forward-coordinate, and complete-head permutations represented by our graph construction, however, PARIS is already invariant by construction. Likewise, the supported query–key and value–output changes of basis are removed by the paired attention encoding before the metanetwork processes the graph. Using these transformations as contrastive augmentations would be unnecessary for enforcing the invariances that our architecture already guarantees. Contrastive augmentation could still be useful for other symmetries that are not built into the current graph representation, but such invariance would be learned empirically.

D.4 Baseline adaptations

Common comparison protocol. All methods operate on the same representation sites, site pairs, and candidate banks. For a comparison rule $h \in \mathcal{H}$, the target score for sites u and v is

$$y_{uv}^{(h)} = \Delta_h(\boldsymbol{H}_u, \boldsymbol{H}_v), \tag{114}$$

where smaller values indicate more similar representations. For retrieval, a query site u ranks candidate sites v in ascending order of either a predicted dissimilarity or a directly measured comparison score. Within each experiment, all methods use the same reference distribution and the same valid

site pairs. CIFAR-10 and CIFAR-100 define separate evaluation settings. Candidates from the query models are excluded.

Information available at test time. Response-based methods evaluate the compared models on a shared set of inputs and use the resulting representation matrices. weightspace methods instead receive only the parameterized upstream computations $\mathcal{C}_u$ and $\mathcal{C}_v$. These computations include the architecture, trained parameters, and evaluation-time state needed to produce the two representations. They exclude downstream weights, model identifiers, source-dataset labels, and any statistics computed from reference responses. During meta-training, learned weightspace methods use target scores computed from reference responses. At test time, they do not need reference inputs or base-model forward passes.

Shared training protocol. All learned comparators use the same model-disjoint splits, sampled site pairs, and target scores. We train them with the regression and ranking losses from the main text. Independent encoders also use the same embedding-variance regularizer. We select models by the same mean-correlation criterion on meta-validation models. We compute normalization statistics and any self-supervised pretraining only from meta-training models. We tune baseline-specific hyperparameters on meta-validation models without using test targets.

Common head for independent encoders. For each independent baseline b, we map its observation of $\mathcal{C}_u$ to a site embedding

$$\boldsymbol{z}_u^{(b)} = E_b\left(\mathrm{Obs}_b(\mathcal{C}_u)\right) \in \mathbb{R}^k. \tag{115}$$

For comparison rule h, we then compute

$$d_{b,h}(u,v) = \left\| \boldsymbol{W}_{b,h}\left(\boldsymbol{z}_u^{(b)} - \boldsymbol{z}_v^{(b)}\right)\right\|_2 \tag{116}$$

and calibrate this distance to the target scale, $\widehat{y}_{uv,b}^{(h)} = q_{b,h}\left(d_{b,h}(u,v)\right)$. We use the same embedding dimension and metric-head family for all independent learned methods, but each method has its own parameters. The calibration is monotone, so it does not change candidate rankings. For retrieval, we use the uncalibrated distance $d_{b,h}$, which also avoids numerical ties when the calibration saturates.

These common heads make the comparison focus on the encoder: each method receives the same supervision and predicts the same target scores, but it represents the upstream computation differently. The heads and supervised objectives are benchmark adaptations and are not part of the original applications of every cited method.

Response-based comparison methods. Response-based baselines evaluate the compared sites on a shared comparison set $\boldsymbol{X}_{\mathrm{cmp}}$. We keep $\boldsymbol{X}_{\mathrm{cmp}}$ disjoint from the reference examples used to define the final evaluation targets. This avoids giving these baselines the responses from which the target score is computed.

1. **SVCCA.** Singular Vector Canonical Correlation Analysis (SVCCA) (Raghu et al., 2017) first reduces each representation by singular value decomposition and then applies canonical correlation analysis (CCA). We evaluate $\boldsymbol{H}_u(\boldsymbol{X}_{\mathrm{cmp}})$ and $\boldsymbol{H}_v(\boldsymbol{X}_{\mathrm{cmp}})$ using the same extraction rules as in the main benchmark. After centering each feature over examples, we retain principal directions that explain 99% of the variance and compute their canonical correlations $\rho_1, \ldots, \rho_r$. We rank candidates by
$$d_{\mathrm{SVCCA}}(u,v) = 1 - \frac{1}{r}\sum_{i=1}^{r}\rho_i. \tag{117}$$
We fix the numerical rank tolerance and CCA regularization on meta-validation data. SVCCA does not use a learned weightspace encoder or a comparison-rule-specific head.
2. **PWCCA.** Projection-Weighted CCA (PWCCA) (Morcos et al., 2018) weights the canonical correlations by how strongly their canonical directions contribute to the original representation. We compute CCA on centered responses and obtain normalized projection weights from the orthonormalized canonical variates. Because the original weighting is directional, we compute the weights in both directions. Let $\alpha_i^{u|v}$ and $\alpha_i^{v|u}$ denote the two sets of weights, each normalized to sum to one. We use
$$d_{\mathrm{PWCCA}}(u,v) = 1 - \frac{1}{2}\left(\sum_i \alpha_i^{u|v}\rho_i + \sum_i \alpha_i^{v|u}\rho_i\right). \tag{118}$$

This symmetrization is an evaluation convention.

Our benchmark compares selected internal sites. A whole-network embedding reused at every site cannot distinguish different locations in the same model, while an isolated layer tensor can omit upstream information that changes the representation. For example, $\boldsymbol{r}_u(\boldsymbol{x}) = \boldsymbol{W}_2\sigma(\boldsymbol{W}_1\boldsymbol{x})$ depends on $\boldsymbol{W}_1$ even when $\boldsymbol{W}_2$ is fixed. We then give learned weightspace baselines the upstream computation $\mathcal{C}_u$ and mark the selected site when the method requires it.

Independent weightspace encoders.

1. **Metadata.** This architecture-only control maps a fixed-length descriptor of $\mathcal{C}_u$ to a site embedding with an MLP. The descriptor contains upstream computational depth, counts of operator types, the target tensor shape, and the number of upstream parameters. We standardize numerical features using meta-training statistics. We do not normalize site depth by the full model depth because this would reveal information downstream of the site. This baseline uses no trained weight values and tests how much of the target score can be predicted from architecture and site location alone.
2. **Tensor marginal.** This control summarizes the marginal distribution of values in each upstream parameter tensor, following the general idea of weightstatistics predictors (Unterthiner et al., 2020). For each tensor, we compute its mean, standard deviation, root-mean-square magnitude, and fixed quantiles, and append descriptors for tensor role and shape. We pool these summaries within tensor-role groups using fixed sums, means, and maxima. An MLP maps the pooled summary, together with the Metadata descriptor, to the common embedding space. This baseline keeps coarse information about parameter distributions but discards coordinate assignments within tensors and correspondences across tensors.
3. **DeepSets.** We adapt the permutation-invariant architecture of Zaheer et al. (2017) to a set of upstream weight tokens. We split tensors into fixed-width blocks, mask padding, and append the tensor type, shape, and position in the upstream computation to each token. If $\xi_{u,j}$ denotes an attributed token, we compute
$$\boldsymbol{z}_u^{\mathrm{DS}} = \rho_{\mathrm{DS}}\left(\left[\sum_j \varphi_{\mathrm{DS}}(\xi_{u,j}),\ \frac{1}{N_u}\sum_j \varphi_{\mathrm{DS}}(\xi_{u,j}),\ \boldsymbol{m}_u\right]\right), \tag{119}$$
where $\boldsymbol{m}_u$ is the Metadata descriptor. Unlike Tensor marginal, DeepSets learns features directly from raw weight tokens. It does not use graph message passing or interactions between tokens. Its permutation invariance concerns the order in which attributed tokens are listed. Note that it does not make the encoder invariant to arbitrary reparameterizations of the network.
4. **SANE-FT.** SANE (Schürholt et al., 2024) represents model weights as fixed-width tokens processed by a Transformer-based sequential autoencoder. For our site-level task, we tokenize only $\mathcal{C}_u$, retain layer and within-layer position information, and restrict all context windows to this upstream computation. We pool the token embeddings with masks and project the result to the common site-embedding space. We pretrain on meta-training sites using SANE's reconstruction and contrastive objectives, discard the decoder, and fine-tune the encoder with the shared comparison-score supervision. We call this adaptation *SANE-FT*. Our heterogeneous-prefix setting does not use whole-network canonical alignment, so it should not be read as a reproduction of the aligned SANE configuration. When we report an aligned variant separately, its reference and all alignment statistics come only from meta-training lineages.
5. **ProbeX-Prefix.** ProbeX (Horwitz et al., 2025) applies learned probes to weight matrices and combines their responses with factorized encoders. Its original formulation can combine information from multiple layers. For our task, we reshape compatible upstream tensors into matrices, apply ProbeX modules, and pool their fixed-dimensional outputs together with tensor-role and prefix-position descriptors. This produces one embedding for the selected site, which is then compared with the common metric heads. We call this adaptation *ProbeX-Prefix*. ProbeX modules are shared across tensors with compatible shapes. When a matrix shape is unsupported, we report the missing coverage and use a labeled matched-subset comparison. The probes act on weights, not on CIFAR inputs, and the method receives no lineage or model-identity labels.
6. **Graph Metanetwork.** We adapt the parameter-graph encoder of Lim et al. (2024a) to $\mathcal{C}_u$. We retain its parameter-attributed graph construction and message-passing backbone, including learned edge features. A target marker identifies the selected site. We replace the original task-specific output with a readout that pools features from the upstream graph and the target

nodes and maps them to the common embedding space. We retain the published parameter-graph representation. This baseline isolates the effect of our graph construction under the same site-level supervision and metric-head family.

Graph Matching Network. We adapt the Graph Matching Network (GMN) of Li et al. (2019), which combines within-graph message passing with cross-graph matching. GMN receives the same parameter-attributed site graphs and target markers as the joint PARIS model. Cross-graph matching compares each node state with an attention-weighted summary of the other graph, and gated pooling produces pair-dependent graph summaries. We replace the original task output and loss with H bounded comparison-score predictions trained with the shared regression and ranking losses. When needed, we symmetrize the GMN score by averaging the two input orders. This adaptation evaluates its joint matching architecture under our site-level supervision.

Stitching-selection baselines. The stitching experiment asks whether a comparison score can screen useful connections before adapter fitting. We use frozen-model stitching (Bansal et al., 2021). We do not compare complete model-assembly systems. Instead, every selector receives the same candidate bank, chooses a fixed-size shortlist, and then uses the same adapter-fitting and validation procedure. The experiment isolates candidate-selection quality. This experiment tests whether these scores provide a useful screening signal under a limited adapter-fitting budget.

A stitching candidate $c = (A, s, B, t)$ connects (A, s) to (B, t): $f_c = B_{>t} \circ S_c \circ A_{\leq s}$, where S_c is the trainable adapter and $B_{>t}$ starts after receiver site t. For each selection problem, all methods receive the same eligible candidate set $\mathcal{K}$ and choose k candidates, with $k \leq |\mathcal{K}|$. Eligibility fixes the architecture family, stitching direction, adapter compatibility, and any compute constraint before a selector scores the candidates. Both constituent networks remain frozen, including their evaluation-time normalization state. We fit a separate adapter for every selected candidate using the same initialization and optimization protocol.

Using learned comparison scores for stitching. For candidate $c = (A, s, B, t)$, let $u = (A, s)$ and $v = (B, t)$. We choose the comparison rule $h^\star$ on meta-validation and keep it fixed for test evaluation. An independent learned method ranks candidates by $s_b(c) = d_{b,h^\star}(u, v)$, while a joint method uses $s_b(c) = \widehat{y}^{(h^\star)}_{uv,b}$.

SN-Net-H. SN-Net uses nearest-anchor selection and paired/unpaired block matching to define a stitching space (Pan et al., 2023). Its full method also trains an elastic collection of stitched models, which differs from our frozen-model protocol. We isolate its architectural selection heuristic and call the resulting baseline *SN-Net-H*.

We order anchor configurations by standalone FLOPs at the benchmark input resolution. Independent runs with the same configuration share the same position. Within compatible stages, we use the published paired/unpaired construction, including nearest-neighbor interpolation for unequal depths, with window size two and stride one. A one-block stage uses a singleton window. After intersecting these configurations with $\mathcal{K}$, we obtain a preferred set $\mathcal{K}_H$.

The SN-Net heuristic does not define a unique top-k ranking. We shuffle $\mathcal{K}_H$ with a recorded random seed and take its first k candidates. If $|\mathcal{K}_H| < k$, we keep all preferred candidates and sample the remaining candidates uniformly without replacement from $\mathcal{K} \setminus \mathcal{K}_H$. We report the coverage of $\mathcal{K}_H$ and how often this fallback is used. We reuse the same shuffled order across shortlist sizes to obtain nested lists. This sampling rule and the frozen-model fitting protocol are our benchmark adaptations. They are not part of the original SN-Net training procedure.

KLAS-score. KLAS uses task probes to construct a normalized directional KL score for stitching selection (Sanyal et al., 2026). We evaluate this score within our common candidate bank and call the resulting baseline *KLAS-score*. For every required site, we fit an affine classifier on a fitting set $\mathcal{B}_{\text{fit}}$ while keeping the base model frozen. For site (A, s),

$$\boldsymbol{p}_{A,s}(x) = \operatorname{softmax}\left(\boldsymbol{U}_{A,s}\boldsymbol{r}_{(A,s)}(\boldsymbol{x}) + \boldsymbol{b}_{A,s}\right) \in \mathbb{R}^C, \tag{120}$$

where C is the number of downstream classes. All probes use the same downstream label space. We choose probe hyperparameters on the development model and compute scores on a disjoint ranking set $\mathcal{B}_{\text{rank}}$. Write

$$\Theta\left((A, s), (B, t)\right) = \frac{1}{|\mathcal{B}_{\text{rank}}|} \sum_{x \in \mathcal{B}_{\text{rank}}} D_{\text{KL}}\left(p_{A,s}(x) \,\|\, p_{B,t}(x)\right). \tag{121}$$

For candidate $c = (A, s, B, t)$, KLAS-score ranks by

$$\Gamma(c) = \frac{\Theta\left((A, s), (B, t)\right)}{\max\left\{\Theta\left((B, t), (B, t+1)\right), \epsilon\right\}}, \qquad \epsilon = 10^{-8}. \tag{122}$$

The denominator compares the receiver site with the output after its next block. We keep the directed KL divergence and take a ratio of the sample-averaged divergences.

The full KLAS procedure also performs anchor filtering and threshold-based selection across FLOPs buckets. To compare its score on the same candidate bank as the other methods, we omit the preliminary anchor filter and select the k candidates with the smallest $\Gamma(c)$ inside the prescribed compute interval. Thus, this baseline evaluates the KLAS selection score. KLAS-score retains its response and task-label access, and we do not remove this information to imitate our weights-only setting.

E Additional Experiments

E.1 Assessing Cross-Graph Interaction

The comparison between PARIS-Ind and PARIS-Joint changes both how the site graphs are processed and how their relation is predicted. To examine whether the joint estimator's advantage can be explained by its more flexible comparison head, we evaluate PARIS-Ind-MLP. This variant retains partner-independent site embeddings but replaces the projected Euclidean distance and exponential calibration with a symmetric nonlinear one:

$$\widehat{\boldsymbol{y}}_{uv} = \sigma\left(\mathrm{MLP}\left([\boldsymbol{z}_u + \boldsymbol{z}_v \parallel |\boldsymbol{z}_u - \boldsymbol{z}_v| \parallel \boldsymbol{z}_u \odot \boldsymbol{z}_v]\right)\right), \tag{123}$$

where $\boldsymbol{z}_u$ and $\boldsymbol{z}_v$ are computed independently and σ is an elementwise sigmoid. Thus, the final comparison is no longer restricted to a calibrated projected distance, but there is no cross-graph interaction before site-level pooling. Fig. 9 shows that replacing the metric head improves the reported prediction score. Yet, PARIS-Joint remains ahead of the MLP control. Thus, decoder flexibility contributes to the observed improvement, but simply replacing it can not compete with the joint estimator. The effect of the MLP control is less uniform for retrieval. It improves the reported retrieval criterion in three of the four settings, but the ViT/CIFAR-10 value increases. The advantage of the joint design extends to the downstream correspondence-selection task and not appearing only in aggregate relation-prediction scores.

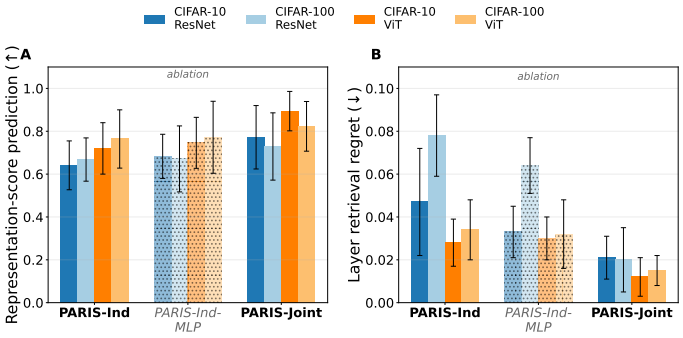

Figure 9: Cross graph interaction analysis. PARIS-Ind-MLP replaces the independent estimator's calibrated metric head with a symmetric MLP while retaining partner-independent site embeddings. The MLP improves prediction scores but does not close the gap to PARIS-Joint.

E.2 Ablating Regression and Ranking Supervision

The supervised objective combines a smooth regression loss $\mathcal{L}_{\mathrm{reg}}^{(h)}$, which fits the numerical target dissimilarities, with an ordinal loss $\mathcal{L}_{\mathrm{rank}}^{(h)}$, which encourages their relative ordering. To assess the contribution of each component, we compare the combined objective with regression only and ranking-only pair supervision for both PARIS-Ind and PARIS-Joint. Fig. 10 reports representation-score prediction across the four dataset-architecture settings. The combined objective achieves the highest reported mean for both estimators in all settings. Regression-only supervision outperforms ranking-only supervision. However, both losses combined still perform the best. Regression directly

constrains the numerical dissimilarities, while ranking explicitly penalizes incorrect relative ordering across supervised site pairs. Although the ranking loss also encourages separation between predictions, it does not directly fit the magnitudes of the target dissimilarities. Conversely, regression alone does not include an explicit pair-ordering objective.

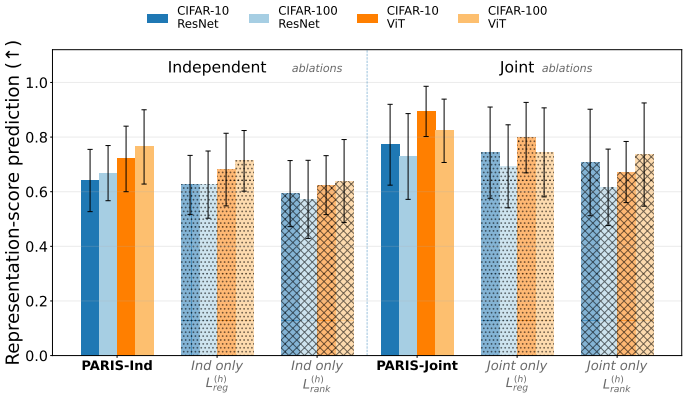

Figure 10: Complementary regression and ranking supervision. Representation-score prediction (↑) for PARIS-Ind and PARIS-Joint using combined regression and ranking pair supervision, regression alone, or ranking alone.

E.3 Sensitivity to the sample size

We examine the effect of the number of inputs in the ordered reference sample used to compute activation-derived dissimilarities. Fig. 11 reports predicted-reference dissimilarity correlations for $n \in \{500, 1000, 2000, 5000\}$ across the four dataset-architecture settings. Increasing the reference size from 500 to 1000 improves the correlation for both estimators. The larger-reference regime has smaller gains than the initial increase to inputs.

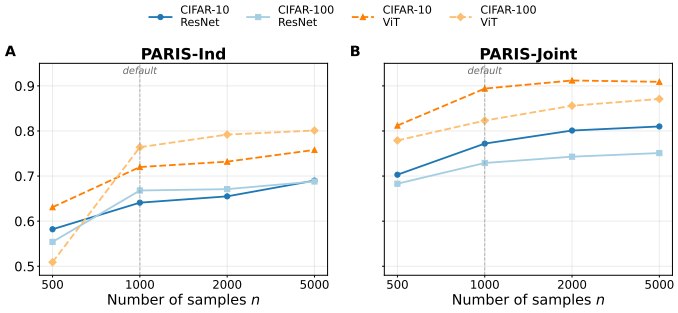

Figure 11: Effect of reference-sample size on predicted-reference correlation. Correlation between predicted and activation-derived reference dissimilarities for PARIS-Ind and PARIS-Joint.

E.4 Per-representation score correlation

Fig. 12 reports the correlations between predicted and corresponding reference dissimilarities. We see that PARIS-Joint yields higher predicted benchmark comparisons. These results indicate that the joint estimator's advantage in correlation extends across several reference internal neural representations. The independent estimator is also competitive and has higher correlation for linear CKA on ResNet/CIFAR-10, and for GULP and MutualNN on ViT/CIFAR-100. The benefit varies with both the representation similarity and model population.

E.5 Meta-training across diverse architectures

We study how the architectures used for meta-training affect the prediction of representation similarity. We compare two settings. In *Same family*, we train a separate metanetwork for each architecture

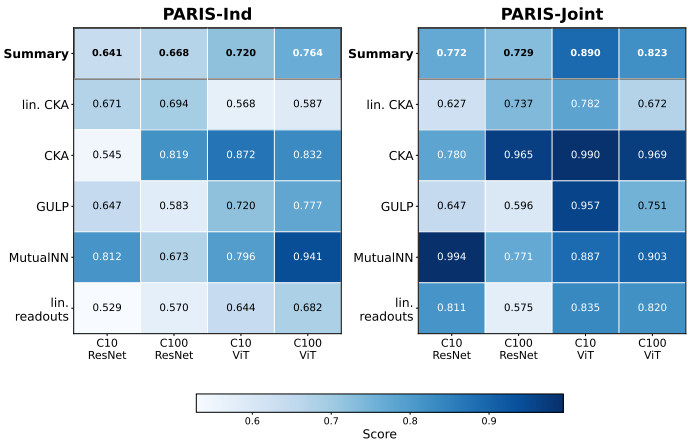

Figure 12: Per-representation score correlation between predicted and reference dissimilarities.

family. In *Diverse*, we train one metanetwork on a shared population that includes ResNets and ViTs. We then evaluate each predictor separately on ResNets and ViTs trained on CIFAR-10 and CIFAR-100. The experiment tests whether a predictor trained on different architecture families can still generalize well to each family.

Fig. 13 shows that both variants continue to track the reference rankings after diverse meta-training. Family-specific training gives higher mean correlations in every setting we evaluate. The gap is larger for ResNets than for ViTs on both datasets. These results suggest that a single metanetwork can predict representation relations across different architecture families, although training within one family gives more accurate predictions in our experiments.

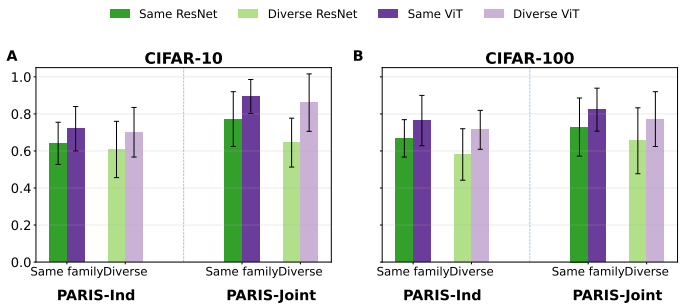

Figure 13: Representation-score prediction under diverse-architecture meta-training. Spearman correlation between predicted and reference representation dissimilarities on CIFAR-10 and CIFAR-100, reported separately for ResNet and ViT evaluation models.

E.6 Cross-task prediction with partially overlapping label spaces

We examine representation-relation prediction between base models trained on related but nonidentical classification tasks. For each parent dataset, let $\mathcal{R}$ be its set of K class labels. We partition $\mathcal{R}$ into three disjoint subsets: task-A-only classes $\mathcal{R}_A$, shared classes $\mathcal{R}_\cap$, and task-B-only classes $\mathcal{R}_B$, with $\big(|\mathcal{R}_A|, |\mathcal{R}_\cap|, |\mathcal{R}_B|\big) = (0.3K, 0.4K, 0.3K)$. The two task label sets are $\mathcal{Y}_A = \mathcal{R}_A \cup \mathcal{R}_\cap$ and $\mathcal{Y}_B = \mathcal{R}_B \cup \mathcal{R}_\cap$. Each task contains 70% of the parent dataset's classes, and the intersection contains 40% of the parent class universe. For CIFAR-10, this gives two seven-class tasks with four shared classes. For CIFAR-100, it gives two 70-class tasks with 40 shared classes. The class partition defines the source tasks. Base-model training data are restricted to the respective task label sets. We evaluate site pairs with one endpoint from an A-trained model and the other from a B-trained model, separately for ResNets and ViTs within each parent dataset. The metanetwork receives the corresponding site graphs and trained state. For each relation, both sites are evaluated on the same ordered reference inputs to construct the activation-derived target. Fig. 14 reports the correlation between predicted and reference dissimilarities across the evaluated site pairs. We see that the ad-

vantage of the joint pipeline in predicted-reference correlation persists when the two base models are trained on different, partially overlapping class sets.

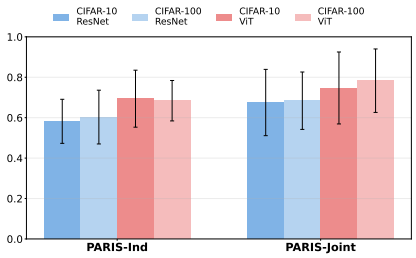

Figure 14: Cross-task correlation under partially overlapping label spaces. We partition each parent dataset's classes into 30% task-A-only, 40% shared, and 30% task-B-only subsets. Each task contains 70% of the original classes. For pairs containing one site from each task-specific base-model population, entries report aggregate correlation between predicted and activation-derived reference dissimilarities.