Connecting Neural Models Latent Geometries with Relative Geodesic Representations

Hanlin Yu[1] University of Helsinki
Berfin Inal University of Amsterdam
Georgios Arvanitidis DTU
Søren Hauberg DTU
Francesco Locatello IST Austria
Marco Fumero[1] IST Austria

Abstract

Neural models learn representations of high-dimensional data on low-dimensional manifolds. Multiple factors, including stochasticities in the training process, model architectures, and additional inductive biases, may induce different representations, even when learning the same task on the same data. However, it has recently been shown that when a latent structure is shared between distinct latent spaces, relative distances between representations can be preserved, up to distortions. Building on this idea, we demonstrate that exploiting the differential-geometric structure of latent spaces of neural models, it is possible to capture *precisely* the transformations between representational spaces trained on similar data distributions. Specifically, we assume that distinct neural models parametrize approximately the same underlying manifold, and introduce a representation based on the *pullback metric* that captures the intrinsic structure of the latent space, while scaling efficiently to large models. We validate experimentally our method on model stitching and retrieval tasks, covering autoencoders and vision foundation discriminative models, across diverse architectures, datasets, and pretraining schemes.

1 Introduction

Neural models learn meaningful representations of high-dimensional data generalizing to many tasks, spanning different data modalities and domains. Recent research reveals that these models often develop similar internal representations when exposed to similar inputs Li et al. [2015], Moschella et al. [2023], Fumero et al. [2024], Kornblith et al. [2019a], a phenomenon that was observed in biological networks Laakso and Cottrell [2000], Haxby et al. [2001]. Remarkably, even when models have different architectures, their internal representations can frequently be aligned through a simple, e.g., orthogonal, transformation Maiorca et al. [2024], Lähner and Moeller [2024a], Moayeri et al. [2023]. This suggests a certain consistency in how neural nets encode information, emphasizing the importance of studying these internal representa-

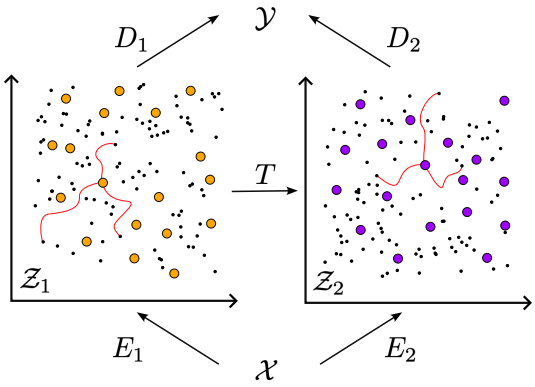


Figure 1: *Neural models trained on similar data learn parametrizations of the same manifold.* NNs learn parametrizations (D_1, D_2) of the same underlying manifold $\mathcal{Y}$ up to isometric transformations $\mathcal{T}$. By pulling back the metric from $\mathcal{Y}$, relative geodesic representations are invariant to the transformation $\mathcal{T}$ between latent spaces $\mathcal{Z}_1$ and $\mathcal{Z}_2$

[1]Corresponding emails: marco.fumero@ist.ac.at, hanlin.yu@helsinki.fi

Preprint. Under review.

tions, and the transformations that relate them, to the extent to hypothesize whether neural nets are converging toward a single representation of reality Huh et al. [2024].

One strategy to understand how different models are related is to identify representations that are *invariant* to transformations between distinct models' representational spaces. A simple and effective recipe is that of *relative representations* Moschella et al. [2023], where samples are represented as a function of a fixed set of latent representations. The similarity function employed is cosine similarity, hinting at the fact that representations across distinct models are subject to *angle preserving* transformations. However, the choice of similarity function should not be limited to only capturing invariances of one class of transformations. As shown in Cannistraci et al. [2024], Fumero et al. [2021], other choices can be good as well, and there's not a clear best choice among different transformations for capturing transformation across distinct latent spaces.We posit that when it is possible to relate distinct neural models' representational spaces, this suggests that neural models are learning distinct parametrizations of the *same* underlying manifold (see Figure 1). In this paper, we employ geodesic distance in the latent space as a metric for relative representations. This approach ensures that the relative space remains approximately invariant to the isometries and reparametrization of the data's manifold, as characterized by a Riemannian structure. Our contributions can be summarized as follows:

- We observe that distinct neural models learn parametrization of the same underlying manifold when trained on similar data.
- We propose a new representation that captures the isometric transformation between data manifolds learned by distinct models, by leveraging the pullback metric.
- We propose to employ a scalable approximation of the geodesic energy to compute intrinsic distances that preserve the ranks of true distances.
- We observe that different pullback metrics are suitable for different tasks, showing for the first time how to get meaningful pullback metrics from discriminative models, such as classifiers and self-supervised models.
- We test relative geodesics on retrieval and stitching tasks on autoencoders and real vision foundation models, across different seeds, architectures, and training strategies, outperforming previous methods.

2 Related Work

Representation alignment. Numerous studies have shown that neural networks trained under different initializations, architectures, or objectives learn highly similar internal feature representations Bonheme and Grzes [2022], Kornblith et al. [2019b], Klabunde et al. [2023], Li et al. [2015], Bengio et al. [2014], Maiorca et al. [2024], Huh et al. [2024], Guth et al. [2024], Chang et al. [2022], Conneau et al. [2018], Tsitsulin et al. [2020], Nejatbakhsh et al. [2024]. This correspondence becomes stronger in wide and large networks Barannikov et al. [2022], Morcos et al. [2018], Somepalli et al. [2022]. Leveraging these aligned embeddings, a simple linear transformation often suffices to map one network's latent space onto another's, enabling techniques such as model stitching, where components from different networks can be interchanged with minimal loss in performance Fumero et al. [2024], Bansal et al. [2021], Csiszárik et al. [2021]. In practice, aligning two independently learned latent spaces often requires only a linear transformation, which achieves comparable downstream task performance [Moayeri et al., 2023, Merullo et al., 2023, Maiorca et al., 2024, Lähner and Moeller, 2024b].

Latent space geometry. Early work on the geometry of deep latent representations focused on autoencoders, where the decoder's mapping from latent to data space induces a natural *pull-back metric* under the assumption that the ambient space is Euclidean [Shao et al., 2018, Tosi et al., 2014, Arvanitidis et al., 2018]. The Riemannian viewpoint allows one to compute geodesic paths and meaningful distances that respect the manifold structure of the learned embedding. Subsequent research has introduced computationally efficient approximations, such as energy-based proxies, and extended these ideas to estimate local curvature for improved interpolation and sampling [Chen et al., 2019, Chadebec and Allassonnière, 2022, Loaiza-Ganem et al., 2024, Arvanitidis et al., 2021, 2022a]. In the context of discriminative models, one can obtain a Riemannian metric primarily using two approaches [Grosse, 2022], either by pulling back the Fisher Information Matrix [Amari, 2016, Arvanitidis et al., 2022b] or by assuming a Euclidean geometry on logit space and pulling back the metric.

3 Method

3.1 Notation and Background

Neural networks (NNs) are parametric functions F_θ, composed of an *encoding* map and a *decoding* map, represented as $F_\theta = D_{\theta_2} \circ E_{\theta_1}$. The encoder $E_{\theta_1} : \mathcal{X} \mapsto \mathcal{Z}$ generates a latent representation $z = E_{\theta_1}(x)$, where $x \in \mathcal{X}$ to the input domain $\mathcal{X}$, and the latent space $\mathcal{Z}$. The decoder D_{θ_2} is responsible for performing the task at hand, such as reconstruction or classification. For simplicity, we omit the parameter dependence (θ) in our notation moving forward. For any single module E (or equivalently D), we use $E_\mathcal{X}$ to denote that the module E was trained on the domain $\mathcal{X}$. In the next sections, we will provide the necessary background to introduce our method.

Latent Space Communication. Given a pair of domains $\mathcal{X}, \mathcal{Y}$, a pair of neural models trained on them $F^1_\mathcal{X}, F^2_\mathcal{Y}$, and a partial correspondence between the domains $\Gamma : \mathcal{A}_\mathcal{X} \mapsto \mathcal{A}_\mathcal{Y}$ where $\mathcal{A}_\mathcal{X} \subset \mathcal{X}$ and $\mathcal{A}_\mathcal{Y} \subset \mathcal{Y}$, the problem of *latent space communication* is the one of finding a full correspondence $\Lambda : E^1(\mathcal{X}) \mapsto E^2(\mathcal{Y})$ between the two domains, from Γ. In a simplified setting, e.g., two models trained with different initialization or architectures on the same data $\mathcal{X} = \mathcal{Y}$ and the correspondence is the identity. When $\mathcal{X} \neq \mathcal{Y}$, the problem becomes multimodal.

Relative representations. The relative representations framework Moschella et al. [2023] provides a straightforward approach to represent each sample in the latent space according to its similarity to a set of fixed training samples, denoted as *anchors*. Representing samples in the latent space as a function of the anchors corresponds to transitioning from an absolute coordinate frame into a *relative* one defined by the anchors and the similarity function. Given a domain $\mathcal{X}$, an encoding function $E_\mathcal{X} : \mathcal{X} \to \mathcal{Z}$, a set of anchors $\mathcal{A}_\mathcal{X} \subset \mathcal{X}$, and a similarity or distance function $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, the *relative representation* for a sample $x \in \mathcal{X}$ is:

$$RR(z; \mathcal{A}_\mathcal{X}, d) = \bigoplus_{a_i \in \mathcal{A}_\mathcal{X}} d(z, E_\mathcal{X}(a_i)),$$

where $z = E_\mathcal{X}(x)$, and $\bigoplus$ denotes row-wise concatenation. In the original method Moschella et al. [2023], d was the cosine similarity. This choice induces a representation invariant to *angle-preserving transformations*. In this work, our focus is to *leverage the intrinsic geometry of latent spaces to employ a metric that captures isometric transformations between data manifolds.*

Latent space geometry. For the latent space of a neural network, it is generally hard to reason about its Riemannian structure. However, it is often easier to assign a Riemannian structure to the output space. As such, one can define a *pullback metric* from the output space to the latent space, which is a standard operation in Riemannian geometry (see ,e.g., Ch.2.4 of Do Carmo and Flaherty Francis [1992]).

Formally, the decoder $D : \mathcal{Z} \mapsto \mathcal{X}$ takes as input a latent representation $z \in \mathcal{Z}$ and outputs x. Given a Riemannian metric defined on x as $G_\mathcal{X}(x)$. Then, the Riemannian metric at z is

$$G_\mathcal{Z}(z) = \left(\frac{\partial x}{\partial z}\right)^\top G_\mathcal{X}(x) \left(\frac{\partial x}{\partial z}\right) = J_z(D)^T G_\mathcal{X}(x) J_z(D),$$

where $J_z(D)$ is the Jacobian of D at z. The metric tensor $G_\mathcal{X}$ is useful to compute quantities such as lengths, angles, and areas on $\mathcal{M}$. Given a smooth curve $\gamma : [a, b] \mapsto \mathcal{M}$, its arc length is

$$L(\gamma) = \int_a^b \sqrt{v(t)^\top G(t) v(t)^\top} \mathrm{d}t. \tag{1}$$

A slight variation of the above functional gives the geodesic energy $\mathcal{E}$ of γ [Arvanitidis et al., 2018, Shao et al., 2018]

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_a^b v(t)^\top G(t) v(t)^\top \mathrm{d}t, \tag{2}$$

where $v(t) = \dot{\gamma}(t)$. Both can be discretized and approximated in practice using finite difference approaches [Yang et al., 2018, Shao et al., 2018]. Geodesic paths minimize both the arc length and the energy, where the latter is usually preferred for numerical stability Hauberg [2025]. The arc length instead has the property of being *invariant* to reparametrizations of the manifold,

Proposition 3.1 ([Do Carmo and Flaherty Francis, 1992]). *Let $\gamma : [a, b] \to \mathcal{M}$ be a smooth curve on a Riemannian manifold $(\mathcal{M}, G)$, and let $\varphi : [\alpha, \beta] \to [a, b]$ be any smooth, strictly increasing reparametrization. Define $\gamma(\tau)' = \gamma\big(\varphi(\tau)\big)$. Then the Riemannian length of γ is unchanged:*

$$L[\gamma'] = \int_{\alpha}^{\beta} \left\| \tfrac{d\gamma'}{d\tau} \right\|_G d\tau = \int_{a}^{b} \|\dot{\gamma}(t)\|_G \, dt = L[\gamma].$$

3.2 Relative geodesics representations

Algorithm 1 Relative Geodesic Representations

Require: Sample $x \in \mathcal{X}$, anchors $\mathcal{A}_{\mathcal{X}}$, encoder E, decoder D, metric $G_{\mathcal{X}}$, steps N, step size Δt, mode $\in \{\texttt{energy}, \texttt{distance}\}$
Ensure: $RR^{geo}(x; \mathcal{A}_{\mathcal{X}})$
$z \leftarrow E(x), \quad RR^{geo} \leftarrow [\,]$
for $a \in \mathcal{A}_{\mathcal{X}}$ **do**
  $z_a \leftarrow E(a), \quad d \leftarrow 0$
  for $j = 1$ to N **do**
    $\gamma_j \leftarrow (1 - \frac{j}{N})z + \frac{j}{N} z_a$
    $\gamma_{j-1} \leftarrow (1 - \frac{j-1}{N})z + \frac{j-1}{N} z_a$
    $v \leftarrow D(\gamma_j) - D(\gamma_{j-1})$
    $G \leftarrow G_{\mathcal{X}}(D(\gamma_j))$
    $s \leftarrow v^\top G v$
    $d \leftarrow d + \Delta t \cdot (\texttt{energy} \Rightarrow \frac{1}{2}s, \ \texttt{distance} \Rightarrow \sqrt{s})$
  end for
  Append d to RR^{geo}
end for
return RR^{geo}

When considering a differential geometry perspective, the problem of latent space communication can be interpreted as finding a transformation between the data manifolds $\mathcal{M}_1, \mathcal{M}_2$ approximated by two neural models F_1, F_2. The relative representation framework captures this transformation implicitly if equipped with the right metric. A natural candidate for this metric is the geodesic distance defined on $\mathcal{M}_1, \mathcal{M}_2$, respectively. This choice makes the relative representations invariant to isometric transformation of the manifolds $\mathcal{M}_1, \mathcal{M}_2$. However, for high-dimensional problems, the high cost of computing the geodesic renders the above methods inappropriate Shao et al. [2018], Chen et al. [2019]. Furthermore, one can argue against directly using the latent geometry induced by deterministic models from a theoretical perspective [Hauberg, 2019], as it may result in undesirable properties, e.g., the resulting geodesics going outside of the data manifold.

We therefore consider using the approximate energy of the straight line (in the Euclidean sense) connecting the representations in the latent space

$$RR^{geo}(z; \mathcal{A}_{\mathcal{X}}) = \bigoplus_{a_i \in \mathcal{A}_{\mathcal{X}}} \mathcal{E}(\tilde{\gamma}(z, E_{\mathcal{X}}(a_i)))$$

where $\tilde{\gamma}(z_1, bz_2) = (1 - \alpha)z_1 + \alpha z_2$ is the convex combination between the points z_1, z_2.

It can be easily seen that for $\bar{\gamma}$ the three quantities are related by the following bounds:

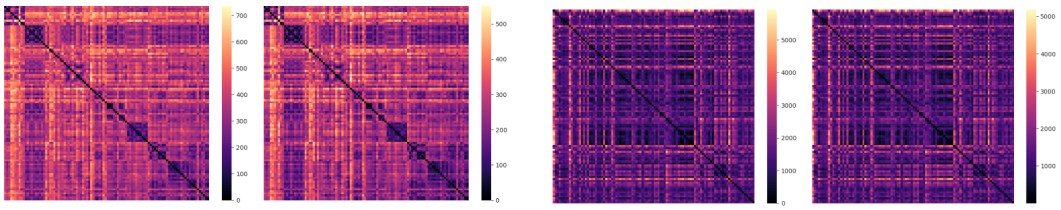
$$d(\mathbf{z}_0, \mathbf{z}_1)^2 \leq L^2(\tilde{\gamma}) \leq 2\mathcal{E}(\tilde{\gamma}) \tag{3}$$

On a Riemannian manifold, a natural choice to form such a representation is to use the Riemannian arc length of a curve defined respectively in Eq. 1 and the energy in Eq. 2.

Discretization. The energy and arc length can be approximated using finite difference schemes,

$$\mathcal{E}(\gamma) = \sum_{i=1}^{N} E_i = \frac{1}{2} \sum_{i=1}^{N} v(t_i)^\top G(t_i) v(t_i) \Delta t, \tag{4}$$

$$L(\gamma) = \sum_{i=1}^{N} d_i = \sum_{i=1}^{N} \sqrt{v(t_i)^\top G(t_i) v(t_i)} \Delta t, \tag{5}$$



(a) MNIST (b) CIFAR-10

Figure 2: Pairwise latent-space distance matrices for (a) MNIST and (b) CIFAR-10. In each subfigure, the left heatmap shows the straight-line energy proxy and the right shows the full Riemannian geodesic distances. The Spearman rank correlation between the two measures is $\rho = 0.99$ for MNIST and $\rho = 1.00$ for CIFAR-10, demonstrating near-perfect agreement.

where $\Delta t = \frac{1}{N}$, with N being the number of discretization steps.

When the step size is small enough, both quantities in the latent space can be approximated by their counterpart on the output space [Shao et al., 2018]. For Euclidean geometry, the geodesic arc length is given in closed form as the geodesics are straight lines. Note that, unlike the energy, the curve length is invariant under reparametrizations (proposition 3.1). As such, we use the curve length.

Approximate geodesic distances. Our choice comes with three advantages (i) Efficiency: since we avoid gradient descent, the computation reduces to a single forward pass for each step in γ, (ii) The approach is minimal yet sufficient as we only need reasonably accurate estimates of the lengths rather than the entire geodesic trajectory, (iii) Since we don't have to perform any optimizations we can use directly the arc length benefiting from its invariance to reparametrizations. To assess how close the straight line energy approximation (2) is to the true geodesic distances, we first encoded 100 samples (10 per class, sorted by label) from MNIST Deng [2012] and CIFAR-10 using a simple autoencoder's encoder. We then computed pairwise distance matrices over these latent representations using both methods, and the results are displayed in Fig. 2. Visually, both distance matrices exhibit the same block-diagonal structure, mainly due to belonging to the same class, and clustering patterns. Numerically, their Spearman rank correlation exceeds 0.99 with only 8 discretization points (see Appendix for correlation results across different numbers of discretization steps and for implementation details).

3.3 Choice of pullback metric

For autoencoders, it has been argued that a pullback metric is beneficial to reflect the underlying geometry of the latent space [Tosi et al., 2014, Arvanitidis et al., 2018, Hauberg, 2019]. For discriminative models, such as classifiers, it is not immediate how to assign a Riemannian structure to the space of latent representations. From the perspective of information geometry, perhaps the most natural choice is the Fisher information matrix [Amari, 2016], in which case the metric in the output space can be obtained as the one with Categorical likelihood. However, neural networks typically experience Neural Collapse [Kothapalli, 2023], rendering the resulting geometry troublesome. Here we discuss two principled approaches: pullback and Diet.

Pullback. Perhaps the most natural idea is, as discussed in Section 3.1, to construct a pullback metric based on the model's outputs. In practice, given a model, we train a classification head upon the latent representations and utilize the resulting Riemannian structure.

Diet. Diet [Ibrahim et al., 2024] is a simple self-supervised training method, based on instance discrimination task, which has been shown to yield identifiability guarantees [Reizinger et al., 2025]. Specifically, it can identify the cluster centers of Von-Mises Fisher (VMF) distributions, which lie on a unit sphere.

One can consider such a scenario [Reizinger et al., 2025]: some latent variables z are drawn from a VMF distribution, which naturally lie on a unit sphere, and pushed forward through an injective generator function g to obtain the data x. Given only x without the knowledge of g, it is possible to

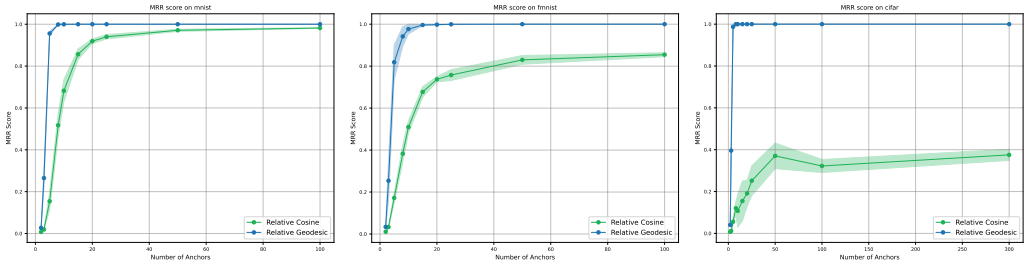


Figure 3: *Aligning latent spaces of autoencoders*: MRR score as a function of the number of anchors on pairs of autoencoders trained with different initializations on the `MNIST` (left), `FashionMNIST` (center), `CIFAR10` (right) datasets, respectively. In green, we plot the performance of Moschella et al. [2023], in blue, our method. The shaded area indicates standard deviation across different random sets of anchors. Relative geodesics consistently outperform the cosine baseline, obtaining peak performance.

recover the latent variables z through parameterizing a model and optimizing

$$L = \mathbb{E}_{x,i}\left[-\log \frac{\exp\left(\langle w_i, f(x)\rangle\right)}{\sum_j \exp\left(\langle w_j, f(x)\rangle\right)}\right],$$

where w is a linear layer without bias and f is a nonlinear encoder. After the model is trained using the above criterion, up to some assumptions, when both f and w are not unit-normalized, $f \circ g$ is linear. This hints that $f \circ g$ may tell us something about the underlying spherical structure of z.

While Diet was proposed to train the entire neural network [Ibrahim et al., 2024], we use it to learn a classification head on top of the pretrained neural network. Specifically, we add several layers on top of the pretrained model, and use it as f. After the model is trained, we may expect $f \circ g$ to give the ground truth z up to a linear transformation. Furthermore, Lähner and Moeller [2024a] noted that it was beneficial to employ data augmentations, which we also observed to be important to achieve good performance.

We propose to directly employ the geodesic distances of a sphere to form the relative representations in Diet. Specifically, on a unit sphere, the distance between two points x and y is given by

$$d(x, y) = \arccos\left(\frac{x^\top y}{\|x\|\,\|y\|}\right),$$

which interestingly bears a strong resemblance to the cosine distances as used in the original paper on relative representations [Moschella et al., 2023]. Observe that x and y do not need to lie precisely on the unit sphere; instead, they are projected onto the unit sphere. As such, this implies the assumption that different instances of $f \circ g$ yield transformations that are constant scalings, which is a stronger assumption than a linear transformation. Moreover, that the data lies on a unit sphere is a rather strong assumption. Nevertheless, as will be shown later, this approach results in meaningful representations across the models.

4 Experimental evaluation on Autoencoders

In the following, we evaluate relative geodesic representations on the latent communication problem across models trained with different initializations, different architectures, and tasks.

4.1 Aligning neural representational spaces trained independently

Experimental setting. For the following experiment we trained pairs of convolutional autoencoders (F_1, F_2) with different initializations on the `MNIST` Deng [2012], `FashionMNIST` Xiao et al. [2017], `CIFAR10` Krizhevsky [2009] datasets. The architecture of the convolutional autoencoder is detailed in the Appendix. After training, we extracted 10k samples from the test set, and mapped them to the latent spaces of the two models, to representations $\mathbf{Z}_1 = E_1(\mathbf{X}), \mathbf{Z}_2 = E_2(\mathbf{X})$ respectively. Starting from a small set of anchors in correspondence $\mathcal{A}_\mathcal{X} \mapsto \mathcal{A}_\mathcal{Y}$, the objective is to evaluate how well it

is possible to recover the full correspondence between the representations $\mathbf{Z}_1, \mathbf{Z}_2$ from the relative representations. As a baseline, we compare with relative representations using cosine similarity Moschella et al. [2023].

Analysis of results. Fig. 3 plots the performance in terms of MRR on `MNIST`, `FashionMNIST`, `CIFAR10` datasets. To obtain the score we first compute similarity matrices between relative representations of the two spaces as $\mathbf{D}(\mathbf{Z}_1, \mathbf{Z}_2)$ where $\mathbf{D}_{i,j} = \frac{RR(\mathbf{Z}_1)_i^T RR(\mathbf{Z}_2)_j}{\|RR(\mathbf{Z}_1)_i\|_2 \|RR(\mathbf{Z}_2)_j\|_2}$. Then we compute the Mean Reciprocal Rank (MRR, see Appendix A.1.1) on top of the similarity matrix. In the figure, we plot MRR as a function of a random set of anchors, where the shaded areas indicate the standard deviations over 5 different sets of random anchors with the same cardinality. Our method consistently performs better than Relative Representation, saturating the score with few anchors on all the domains, despite the different degrees of complexity of the latent spaces. In addition, our method shows significantly less variance, being more robust to the choice of the anchor set.

Takeaway: Relative geodesic representation near-perfectly captures transformations between representational spaces of models initialized differently, outperforming Moschella et al. [2023] in sample efficiency and robustness.

4.2 Stitching autoencoder models

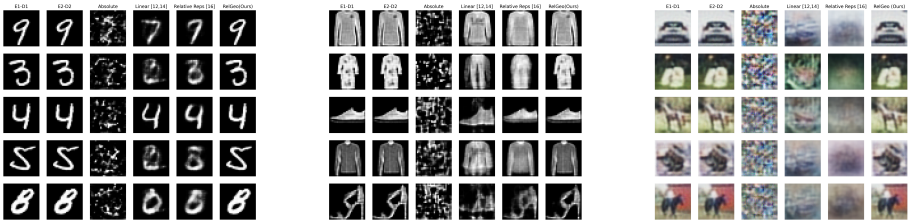


Figure 4: *Stitching on Autoencoders*: We visualize qualitative reconstructions of samples, stitching autoencoders of models trained with different initializations on `MNIST` (left), `FashionMNIST` (center), `CIFAR10` (right). The first two columns show reconstructions from the original models; the middle columns represent baselines Maiorca et al. [2024], Lähner and Moeller [2024a], Moschella et al. [2023]; the rightmost column is our method. Relative geodesics yield the best stitching results using just 5 anchors.

Experimental setting. We consider the same pairs of autoencoders trained on the `MNIST`, `FashionMNIST`, `CIFAR10` datasets of section 4.1. Starting from a set of five random anchors, we estimate a transformation T between the model representational spaces Z_1, Z_2. Different from Moschella et al. [2023], in which zero-shot stitching was achieved by training once a decoder module with relative representations and then exchanging different encoder modules, here we achieve stitching without training any decoder. We compute relative representation with respect to the set of anchors, and compute a similarity matrix $\mathbf{D}(\mathbf{Z}_1, \mathbf{Z}_2)$. Then we compute the vector $\mathbf{c} = \arg\max_i(\mathbf{D})$ representing a correspondence between the two representations matrices $\mathbf{Z}_1$, $\mathbf{Z}_2$, and use c to fit a linear transformation T to approximate the transformation between the two domains. We perform stitching by performing the following operation for a sample $x \in \mathcal{X}$: $\tilde{x} = D_2 \circ T \circ E_1(x)$.

Analysis of results. We visualize the results of reconstructions of random samples in Fig. 4, comparing with Moschella et al. [2023], Lähner and Moeller [2024a], Maiorca et al. [2024]. For each dataset, each column represents respectively: (i) the original autoencoding mapping for a sample x of model F_1, $D_1(E_1(x))$, (ii) $D_2(E_2(x))$, (iii) the mapping $D_2(E_1(x))$, (iv) the mapping $D_2(T_{anchors}E_1(x))$ where $T_{anchors}$ is estimated on the five available anchors, (v) the mapping $D_2(T_{cosine}E_1(x))$ where T_{cosine} is estimated among all 10k samples with the correspondence c obtaining in the relative space of Moschella et al. [2023], (vi) Our result $D_2(T_{relgeo}E_1(x))$, where T_{relgeo} is estimated from the correspondence obtained in the relative geodesic space. While the baselines do not reach a good enough reconstruction quality, reconstructions with our method are almost perfect in accordance with the results in Fig. 3.

Takeaway: The relative geodesic space enables stitching neural modules trained on different seeds.

5 Experiments on vision foundation models

In this section we evaluate relative geodesic representations performance on retrieval and model stitching tasks on vision foundation discriminative models across models trained with different objectives, architectures, and sizes.

5.1 Matching representational spaces of discriminative foundation models

Table 1: Average MRR cosine results for different methods across different datasets. Relative representations pulling back from diet decoder (`RelGeo(Diet)`) consistently provides better retrievals.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
`Rel(Cosine)` Moschella et al. [2023]	0.129 ± 0.135	0.166 ± 0.162	0.221 ± 0.178	0.135 ± 0.148	0.068 ± 0.08
`RelGeo(Pullback)`	0.047 ± 0.013	0.112 ± 0.031	0.412 ± 0.09	0.28 ± 0.129	0.025 ± 0.012
`RelGeo(Diet)`	$\mathbf{0.387} \pm 0.145$	$\mathbf{0.445} \pm 0.142$	$\mathbf{0.566} \pm 0.111$	$\mathbf{0.523} \pm 0.177$	$\mathbf{0.314} \pm 0.188$

In this section, we test the compatibilities of representations of vision foundation models with different architectures, such as residual networks He et al. [2016b] and vision transformers Dosovitskiy et al. [2021], and with different pretraining objectives including classification and self-supervised learning.

Experimental setting. We perform experiments on retrieval tasks on pretrained vision foundation models, investigating how well we can match representations together with different backbones subject to the decoding tasks, on 5 datasets, varying in complexity and size: `CIFAR10`, `CIFAR100` Krizhevsky [2009], `SVHN` Yuval Netzer et al. [2011], `CUB` Wah et al. [2023], and `ImageNet-1k` Russakovsky et al. [2015]. For ImageNet-1k, we used 1000 anchors, while for other datasets we used 500. As backbones we consider ResNet-50 [He et al., 2016a], Vision Transformers (ViT) [Dosovitskiy et al., 2021], with both patch 16-224 and patch 32-384, and DINOV2 [Oquab et al., 2024]. We compare the original formulation of relative representations with cosine similarity Moschella et al. [2023] denoted as `Rel(Cosine)`, relative geodesic representation using Euclidean pullback metric denoted as `RelGeo(Pullback)`, and pulling back the spherical metric using a Diet decoder denoted `RelGeo(Diet)`.

Analysis of results. Table 1 shows results from different methods averaged across all possible pairs of models on the considered datasets. Additionally, Fig. 5 shows the results on CUB datasets, highlighting the numbers in different settings. While `RelGeo(Pullback)` may result in worse MRR numbers, `RelGeo(Diet)` provides consistently improved retrieval performance. In the Appendix we report full results for every dataset.

Takeaway: Relative geodesic representations pulling back from instance discrimination decoders are identifiable across vision foundation models, improving retrieval performances.

5.2 Zero-shot stitching of vision foundation models

Table 2: Average stitching performances across different settings. `RelGeo(Pullback)` often outperforms `Rel(Cosine)`, while `RelGeo(Diet)` remains competitive.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
`Rel(Cosine)` Moschella et al. [2023]	0.907 ± 0.09	0.775 ± 0.132	$\mathbf{0.549} \pm 0.152$	0.531 ± 0.188	0.384 ± 0.115
`RelGeo(Pullback)`	$\mathbf{0.955} \pm 0.03$	$\mathbf{0.874} \pm 0.055$	0.501 ± 0.159	$\mathbf{0.595} \pm 0.163$	$\mathbf{0.59} \pm 0.054$
`RelGeo(Diet)`	0.915 ± 0.074	0.775 ± 0.115	0.479 ± 0.17	0.559 ± 0.171	0.416 ± 0.079

Model stitching was introduced in Lenc and Vedaldi [2015] to analyze neural network representational spaces, by training a linear layer to connect different layers and evaluating performance. Here we sidestep the need for trainable stitching layers and consider the zero-shot model stitching task defined in Moschella et al. [2023] to effectively test how components of vision foundation models can be reused. To do this, we leverage the space of relative geodesic representations as a shared compatible space. For the ith model E_i, we train one decoder D_i on the relative representations induced by it, then evaluate the performance of using D_i to decode the representations of model E_j, where E_j may be a different model. This assesses how much two representation spaces can be merged with respect to the task defined by decoder D, e.g., a classification head.

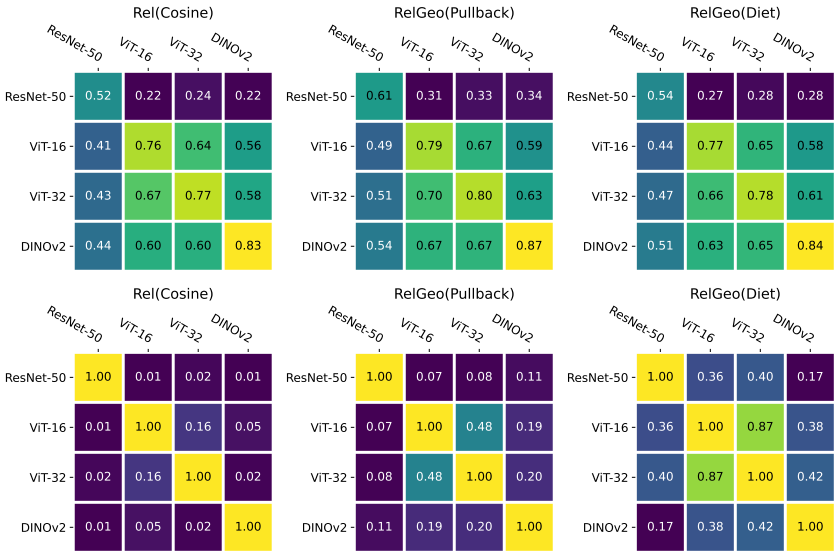


Figure 5: CUB Accuracies (top) and symmetricized MRR cosine (bottom)

Experimental setting. We perform experiments on pretrained vision foundation models from Hugging Face, investigating how well we can match representations together with different backbones with classification heads, on the same datasets and models as considered in Section 5.1. Similar to Section 5.1, we compare `Rel(Cosine)`, `RelGeo(Pullback)` and `RelGeo(Diet)`.

Analysis of results. The results of the different methods across the different data sets are shown in Table 2, where we average over all possible model pairs. We further show the accuracies of the models on the CUB dataset in Fig. 5. Both `RelGeo(Pullback)` and `RelGeo(Diet)` provide strong stitching accuracies, with `RelGeo(Pullback)` reflecting the benefits of pulling back class specific information. `RelGeo(Diet)` results still in good accuracies while having very strong MRR metrics 1.

Takeaway: Using geometric relative representations yields good accuracies and good MRRs, avoiding downgrading of performance when performing model stitching while retaining identifiability.

6 Conclusions and discussion

We have introduced the framework of relative geodesic representation starting from the assumption that distinct neural models trained on similar data distributions learn to approximate the same underlying latent manifold. As a result, geodesic distances based on their representations are invariant to transformations between different representational spaces. We show that the geodesic energy and arc length of straight lines provide an efficient, low-cost metric for bridging these spaces, allowing us to measure similarity and align representations across different architectures, training objectives, and training procedures, while outperforming previous methods.

Limitations and future work. The accuracy of using the straight line arc length (or energy) approximation can be imprecise in regions of high curvature in the latent space, corresponding to regions further from the training points' support. Moreover, this could require increasingly smaller step sizes, hurting the efficiency performance of the method. This suggests considering different paths rather than the linear one, and adaptive step sizes, e.g., by estimating the support of the data building KNN graphs in the latent space and forcing the path to not deviate too much from them. By employing the pullback metric from a given output space, the relative geodesics representation has the interesting property of restricting the alignment problem to the information relevant to the decoding task. This could be useful to (i) explore multi-modal alignment Norelli et al. [2023], where it is of interest to capture not only the shared information across modalities, but also the modality-specific information; (ii) to better understand the relation between the representation similarity and decodability Harvey et al. [2024] and the interaction between tasks and learned representations Fumero et al. [2023].

Acknowledgments and Disclosure of Funding

We thank Gregor Krzmanc, German Magai, Vital Fernandez for insightful discussions in the early stages of the project. HY was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI. HY wishes to acknowledge CSC - IT Center for Science, Finland, for computational resources. GA was supported by the DFF Sapere Aude Starting Grant "GADL". SH was supported by a research grant (42062) from VILLUM FONDEN and partly funded by the Novo Nordisk Foundation through the Center for Basic Research in Life Science (NNF20OC0062606). SH received funding from the European Research Council (ERC) under the European Union's Horizon Programme (grant agreement 101125003). MF is supported by the MSCA IST-Bridge fellowship which has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101034413.

References

S.-i. Amari. *Information Geometry and Its Applications*. Springer Tokyo, 1 edition, 2016.

G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent Space Oddity: on the Curvature of Deep Generative Models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJzRZ-WCZ.

G. Arvanitidis, S. Hauberg, and B. Schölkopf. Geometrically Enriched Latent Spaces. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

G. Arvanitidis, B. Georgiev, and B. Schölkopf. A prior-based approximate latent Riemannian metric. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022a.

G. Arvanitidis, M. González-Duque, A. Pouplin, D. Kalatzis, and S. Hauberg. Pulling back information geometry. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4872–4894. PMLR, Mar. 2022b.

Y. Bansal, P. Nakkiran, and B. Barak. Revisiting model stitching to compare neural representations. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 225–236. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/01ded4259d101feb739b06c399e9cd9c-Paper.pdf.

S. Barannikov, I. Trofimov, N. Balabin, and E. 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*, pages 1607–1626. PMLR, 17–23 Jul 2022.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives, 2014. URL https://arxiv.org/abs/1206.5538.

L. Bonheme and M. Grzes. How do variational autoencoders learn? insights from representational similarity, 2022. URL https://arxiv.org/abs/2205.08399.

I. Cannistraci, L. Moschella, M. Fumero, V. Maiorca, and E. Rodolà. From bricks to bridges: Product of invariances to enhance latent space communication. *ICLR*, 2024.

C. Chadebec and S. Allassonnière. A geometric perspective on variational autoencoders, 2022. URL https://arxiv.org/abs/2209.07370.

T. A. Chang, Z. Tu, and B. K. Bergen. The geometry of multilingual language model representations, 2022. URL https://arxiv.org/abs/2205.10964.

N. Chen, F. Ferroni, A. Klushyn, A. Paraschos, J. Bayer, and P. van der Smagt. Fast approximate geodesics for deep generative models. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Deep Learning: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part II 28*, pages 554–566. Springer, 2019.

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data, 2018. URL https://arxiv.org/abs/1710.04087.

A. Csiszárik, P. Kőrösi-Szabó, Ákos K. Matszangosz, G. Papp, and D. Varga. Similarity and matching of neural network representations, 2021. URL https://arxiv.org/abs/2110.14633.

L. Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

N. S. Detlefsen, A. Pouplin, C. W. Feldager, C. Geng, D. Kalatzis, H. Hauschultz, M. González-Duque, F. Warburg, M. Miani, and S. Hauberg. Stochman. *GitHub. Note: https://github.com/MachineLearningLifeScience/stochman/*, 2021.

M. P. Do Carmo and J. Flaherty Francis. *Riemannian geometry*, volume 2. Springer, 1992.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

M. Fumero, L. Cosmo, S. Melzi, and E. Rodolà. Learning disentangled representations via product manifold projection. In *International conference on machine learning*, pages 3530–3540. PMLR, 2021.

M. Fumero, F. Wenzel, L. Zancato, A. Achille, E. Rodolà, S. Soatto, B. Schölkopf, and F. Locatello. Leveraging sparse and shared feature activations for disentangled representation learning. *Advances in Neural Information Processing Systems*, 36:27682–27698, 2023.

M. Fumero, M. Pegoraro, V. Maiorca, F. Locatello, and E. Rodolà. Latent functional maps: a spectral framework for representation alignment. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 66178–66203. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/79be41d858841037987964e3f5caf76d-Paper-Conference.pdf.

R. Grosse. Chapter 3: Metrics, 2022. URL https://www.cs.toronto.edu/~rgrosse/courses/csc2541_2022/readings/L03_metrics.pdf.

F. Guth, B. Ménard, G. Rochette, and S. Mallat. A rainbow in deep network black boxes, 2024. URL https://arxiv.org/abs/2305.18512.

S. E. Harvey, D. Lipshutz, and A. H. Williams. What representational similarity measures imply about decodable information. *arXiv preprint arXiv:2411.08197*, 2024.

S. Hauberg. Only Bayes should learn a manifold (on the estimation of differential geometric structure from data), 2019. _eprint: 1806.04994.

S. Hauberg. *Differential geometry for generative modeling*. DTU, 2025.

J. V. Haxby, M. I. Gobbini, M. L. Furey, A. Ishai, J. L. Schouten, and P. Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539): 2425–2430, 2001.

K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016a. doi: 10.1109/CVPR.2016.90.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016b.

M. Huh, B. Cheung, T. Wang, and P. Isola. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

M. Ibrahim, D. Klindt, and R. Balestriero. Occam's Razor for Self Supervised Learning: What is Sufficient to Learn Good Representations?, 2024. URL https://arxiv.org/abs/2406.10743. _eprint: 2406.10743.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

M. Klabunde, T. Schumacher, M. Strohmaier, and F. Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *arXiv preprint arXiv:2305.06329*, 2023.

S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019a.

S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR, 09–15 Jun 2019b. URL https://proceedings.mlr.press/v97/kornblith19a.html.

V. Kothapalli. Neural Collapse: A Review on Modelling Principles and Generalization. *Transactions on Machine Learning Research*, 2023.

A. Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, pages 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

A. Laakso and G. Cottrell. Content and cluster analysis: Assessing representational similarity in neural systems. *Philosophical Psychology*, 13:47 – 76, 2000.

Z. Lähner and M. Moeller. On the direct alignment of latent spaces. In *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, volume 243 of *Proceedings of Machine Learning Research*, pages 158–169. PMLR, 2024a. URL https://proceedings.mlr.press/v243/lahner24a.html.

Z. Lähner and M. Moeller. On the direct alignment of latent spaces. In M. Fumero, E. Rodolá, C. Domine, F. Locatello, K. Dziugaite, and C. Mathilde, editors, *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*, volume 243 of *Proceedings of Machine Learning Research*, pages 158–169. PMLR, 15 Dec 2024b. URL https://proceedings.mlr.press/v243/lahner24a.html.

K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence, 2015. URL https://arxiv.org/abs/1411.5908.

Q. Lhoest, A. Villanova del Moral, P. von Platen, T. Wolf, M. Šaško, Y. Jernite, A. Thakur, L. Tunstall, S. Patil, M. Drame, J. Chaumond, J. Plu, J. Davison, S. Brandeis, V. Sanh, T. Le Scao, K. Canwen Xu, N. Patry, S. Liu, A. McMillan-Major, P. Schmid, S. Gugger, N. Raw, S. Lesage, A. Lozhkov, M. Carrigan, T. Matussière, L. von Werra, L. Debut, S. Bekman, and C. Delangue. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184. Association for Computational Linguistics, Nov. 2021.

Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

G. Loaiza-Ganem, B. L. Ross, R. Hosseinzadeh, A. L. Caterini, and J. C. Cresswell. Deep generative models through the lens of the manifold hypothesis: A survey and new connections, 2024. URL https://arxiv.org/abs/2404.02954.

V. Maiorca, L. Moschella, A. Norelli, M. Fumero, F. Locatello, and E. Rodolà. Latent space translation via semantic alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

J. Merullo, L. Castricato, C. Eickhoff, and E. Pavlick. Linearly mapping from image to text space, 2023. URL https://arxiv.org/abs/2209.15162.

M. Moayeri, K. Rezaei, M. Sanjabi, and S. Feizi. Text-to-concept (and back) via cross-model alignment. In *International Conference on Machine Learning*, pages 25037–25060. PMLR, 2023.

A. S. Morcos, M. Raghu, and S. Bengio. Insights on representational similarity in neural networks with canonical correlation, 2018. URL https://arxiv.org/abs/1806.05759.

L. Moschella, V. Maiorca, M. Fumero, A. Norelli, F. Locatello, and E. Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SrC-nwieGJ.

A. Nejatbakhsh, V. Geadah, A. H. Williams, and D. Lipshutz. Comparing noisy neural population dynamics using optimal transport distances. *arXiv preprint arXiv:2412.14421*, 2024.

A. Norelli, M. Fumero, V. Maiorca, L. Moschella, E. Rodolà, and F. Locatello. Asif: Coupled data turns unimodal models to multimodal without training, 2023. URL https://arxiv.org/abs/2210.01738.

M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

P. Reizinger, A. Bizeul, A. Juhos, J. E. Vogt, R. Balestriero, W. Brendel, and D. Klindt. Cross-Entropy Is All You Need To Invert the Data Generating Process. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=hrqNOxpItr.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

H. Shao, A. Kumar, and P. T. Fletcher. The Riemannian Geometry of Deep Generative Models. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 428–4288, 2018. doi: 10.1109/CVPRW.2018.00071.

G. Somepalli, L. Fowl, A. Bansal, P. Yeh-Chiang, Y. Dar, R. Baraniuk, M. Goldblum, and T. Goldstein. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective, 2022. URL https://arxiv.org/abs/2203.08124.

A. Tosi, S. Hauberg, A. Vellido, and N. D. Lawrence. Metrics for Probabilistic Geometries. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 800–808, Arlington, Virginia, USA, 2014. AUAI Press. event-place: Quebec City, Quebec, Canada.

A. Tsitsulin, M. Munkhoeva, D. Mottin, P. Karras, A. Bronstein, I. Oseledets, and E. Müller. The shape of data: Intrinsic distance for data distributions, 2020. URL https://arxiv.org/abs/1905.11141.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset, Aug. 2023.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *ACL*, pages 38–45. Association for Computational Linguistics, Oct. 2020.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

T. Yang, G. Arvanitidis, D. Fu, X. Li, and S. Hauberg. Geodesic clustering in deep generative models. In *arXiv preprint*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Dataset*, 2011.

A Appendix

A.1 Additional details

A.1.1 Mean Reciprocal Rank

Mean Reciprocal Rank (MRR) is a commonly used metric to evaluate the performance of retrieval systems [Moschella et al., 2023]. It measures the effectiveness of a system by calculating the rank of the first relevant item in the search results for each query.

To compute MRR, we consider the following steps:

1. For each query, rank the list of retrieved items based on their relevance to the query.
2. Determine the rank position of the first relevant item in the list. If the first relevant item for query i is found at rank position r_i, then the reciprocal rank for that query is $\frac{1}{r_i}$.
3. Calculate the mean of the reciprocal ranks over all queries. If there are Q queries, the MRR is given by:

$$\text{MRR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{r_i} \tag{6}$$

   Here, r_i is the rank position of the first relevant item for the i-th query. If a query has no relevant items in the retrieved list, its reciprocal rank is considered to be zero.

MRR provides a single metric that reflects the average performance of the retrieval system, with higher MRR values indicating better performance.

Similar to stitching accuracies, MRR is generally asymmetric. However, it can also be made symmetric. Specifically, as MRR is calculated based on a distance matrix D, one can make the distance matrix symmetric by setting $D = \frac{1}{2}\left(D^\top + D\right)$. In Section 5.1 we reported the symmetric version. Otherwise we report both the original version and the symmetric version, and discriminate between these two by explicitly indicating it when it is symmetric.

A.1.2 Architectural details

We provide here the architectural details of the convolutional Autoencoders employed in experiments in Figures 3 and 4.

Encoder
3×3 conv. 32 stride 2-ReLu
3×3 conv. 64 stride 2-ReLu
Flatten
$(64 * k * k) \times h$ Linear
Latents
Decoder
$h \times (64 * k * k)$ Linear
Unflatten
3×3 conv. 64 stride 2-ReLu
3×3 conv. 32 stride 2-ReLu
Sigmoid

Table 3

For the classifier experiment, in order to obtain geometric representations we need a decoder. The architecture is shown in Table 4. For RelGeo(Diet), the last linear layer is configured with *bias=False* in accordance with the original algorithm.

For evaluating the performances of the representations, we train a classification head with the same architecture as used by Moschella et al. [2023] as given in Table 5.

Classification head
$input\_dim$ LayerNorm
$input\_dim \times 500$ Linear-Tanh
$500 \times num\_classes$ Linear

Table 4

Final classification head
$input\_dim$ LayerNorm
$input\_dim \times input\_dim$ Linear-Tanh
InstanceNorm1d
$input\_dim \times num\_classes$ Linear

Table 5

A.1.3 RelGeo(Diet) augmentations

As noted by Ibrahim et al. [2024], it is beneficial to employ data augmentations when using Diet to perform self-supervised training of neural networks. We largely follow their approach, and considered different levels of data augmentations. Following Ibrahim et al. [2024], we consider different levels of data augmentations indexed by a scalar strength, which are summarized below using PyTorch pseudocode; strengths of a higher level employs the augmentations of lower levels as well.

0: no augmentations;

1: RandomResizedCrop((height, width)), RandomHorizontalFlip();

2: RandomApply(ColorJitter(0.4, 0.4, 0.4, 0.2)), p=0.3); RandomGrayscale(0.2);

3: RandomApply(GaussianBlur((3, 3), (1.0, 2.0)), p=0.2), RandomErasing(0.25).

A.1.4 Compute resources

Experiments regarding the geodesic approximation are conducted using NVIDIA A100 GPU and 12 CPU cores. Run time varies depending on the discretization steps, number of anchors and the used dataset.

The autoencoder stitching and retrieval experiments are performed on an NVIDIA RTX 3080TI GPU.

The experiments concerning vision foundation models are carried out on a compute cluster, with each job using a single NVIDIA A100 GPU and 10 CPU cores and taking several hours.

Preliminary experiments used up more compute resources. It is estimated that we used hundreds of GPU hours.

A.1.5 Geodesic approximation

Here, we provide the experimental details of the results presented in Fig. 2 and Fig. 6. To assess the geodesic energies, we used a small autoencoder, whose architecture is presented in Table 6.

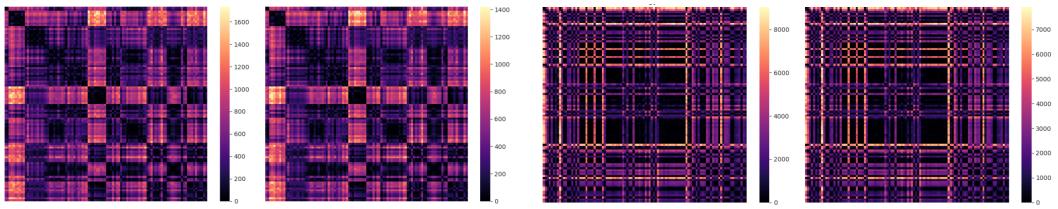
Autoencoder training We trained a lightweight convolutional autoencoder (see Table 6) on both MNIST and CIFAR-10 to obtain the latent representations used in our experiments. For MNIST, the first convolutional layer was adjusted to accept a single input channel; for CIFAR-10 it used three channels. Each model was trained for 30 epochs using the Adam optimizer Kingma and Ba [2017] with a batch size of 64. We set the learning rate to 0.001, and we fixed a random seed of 42 to ensure reproducibility.

Distance computation After training, we selected 10 samples per class (100 total) in label order from each dataset and encoded them to produce their latent encodings. True geodesics are computed using Stochman library Detlefsen et al. [2021], which wraps the decoder into a pull-back manifold,

intializes a spline path between codes, and then optimizes its control points to minimize the Riemannian energy. Geodesic energies are computed as in Eq. 2. Pairwise distances are computed and visualized in Figures 2 and 6, demonstrating the close agreement between the two measures under identical encoding and discretization settings. In Fig. 2, latent dimensions for MNIST and CIFAR are 64 and 128 respectively, while in Fig. 6, latent dimension is 2 for both datasets.

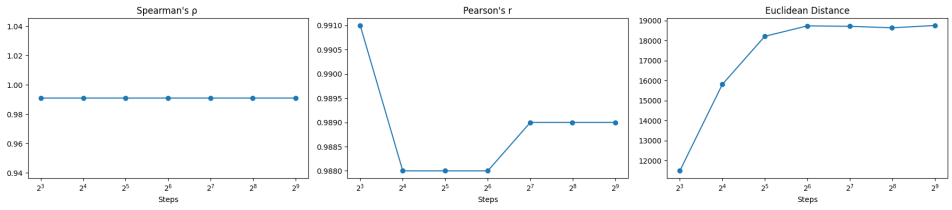
Table 6: ConvAutoencoder architecture (latent dim d).

Encoder	Activation
Conv2d(1, 32, kernel = 3, stride=2, pad=1)	ReLU
Conv2d(32, 64, kernel = 3, stride=2, pad=1)	ReLU
Flatten	—
Linear(64*7*7, d)	—
Decoder	Activation
Linear(d, 64*7*7)	ReLU
Unflatten(64,7,7)	—
ConvTranspose2d(64, 32, kernel = 3, stride=2, pad=1, out_pad=1)	ReLU
ConvTranspose2d(32, 1, kernel = 3, stride=2, pad=1, out_pad=1)	Sigmoid

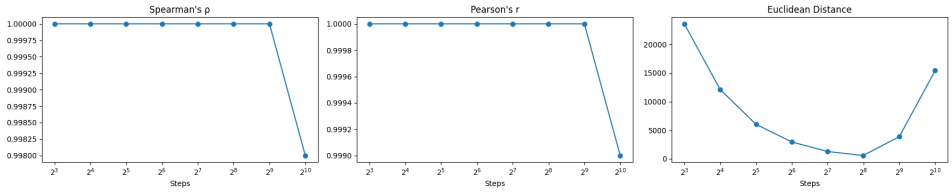


(a) MNIST (b) CIFAR-10

Figure 6: Pairwise latent-space distance matrices for (a) MNIST and (b) CIFAR-10, with latent dimension of 2. In each subfigure, the left heatmap shows the straight-line energy proxy and the right shows the full Riemannian geodesic distances. The Spearman rank correlation between the two measures is 0.99 for MNIST and $\rho = 1.00$ for CIFAR-10, demonstrating near-perfect agreement.



(a) MNIST



(b) CIFAR-10

Figure 7: Impact of varying discretization levels on similarity and distance metrics for (a) MNIST and (b) CIFAR-10 datasets. Each subplot shows how Spearman's ρ, Pearson's r, and Euclidean distance change as the number of discretization levels increases.

Table 7: MRR based on cosine.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
`Rel(Cosine)` Moschella et al. [2023]	0.08 ± 0.077	0.122 ± 0.109	0.21 ± 0.149	0.089 ± 0.094	0.035 ± 0.034
`RelGeo(Pullback)`	0.019 ± 0.005	0.046 ± 0.016	0.236 ± 0.08	0.156 ± 0.089	0.013 ± 0.005
`RelGeo(Diet)`	$\mathbf{0.189} \pm 0.108$	$\mathbf{0.241} \pm 0.117$	$\mathbf{0.358} \pm 0.126$	$\mathbf{0.327} \pm 0.184$	$\mathbf{0.131} \pm 0.107$

A.1.6 Autoencoder stitching and retrieval

We provide the experimental details of the results presented in Figure 3 and Figure 4. All models employed followed the architecture depicted in Table 6, with a latent dimensionality of 128.

We trained the lightweight convolutional autoencoder (see Table 6) on MNIST, CIFAR-10, FashionMNIST with 5 different seeds, to obtain the latent representations used in our experiments. For MNIST, and FashionMNIST the first convolutional layer was adjusted to accept a single input channel; for CIFAR-10 it used three channels. Each model was trained for 50 epochs, reaching convergence, using the Adam optimizer Kingma and Ba [2017] with a batch size of 64. We set the learning rate to 0.001.

A.1.7 Vision foundation models

We use the pretrained models as provided by Huggingface Transformers [Wolf et al., 2020], which has Apache-2.0 license, and the datasets as provided by HuggingFace Datasets [Lhoest et al., 2021], which also has Apache-2.0 license. The license information of the datasets are: CIFAR-10: unknown; CIFAR-100: unknown; CUB: unknown; ImageNet-1k: ImageNet agreement; SVHN: non-commercial use only.

Unless otherwise stated, we directly use the original test set of the dataset as the test set, while using 0.9 of the original train set as the train set and the remaining as the validation set. Both the anchors and the Diet data points are selected from the validation set.

For CIFAR-100, we use coarse labels.

For SVHN, the objective is to predict the cropped digits.

For CUB dataset, we use `https://huggingface.co/datasets/birder-project/CUB_200_2011-WDS`. The training set is of a small size, and we select 2000 points as the validation set.

When reporting aggregated MRR metrics in the tables, we always ignore the diagonal numbers as these are generally (close to) 1.

For all cases where we need to train classification heads, apart from the ones with Diet the heads are trained for 10 epochs, while the ones with Diet are trained for 50 epochs. The heads used to obtain the gometric information are trained using learning rate $5e-4$ and batch size 64, while the heads used for stitching was trained using learning rate $1e-4$ and batch size 32.

When reporting stitching results, we train three classification heads and average the accuracies as the final results.

A.2 Additional results on Vision Foundation models

We provide additional results on vision foundation models. For ablation studies, we focus on the performances of the models on CUB dataset.

A.2.1 Other models

We provide the heatmaps on different datasets in Figure 8, Figure 9, Figure 10 and Figure 11.

A.2.2 Other evaluation metrics

We provide the results of other evaluation metrics in Table 7, Table 8 and Table 9.

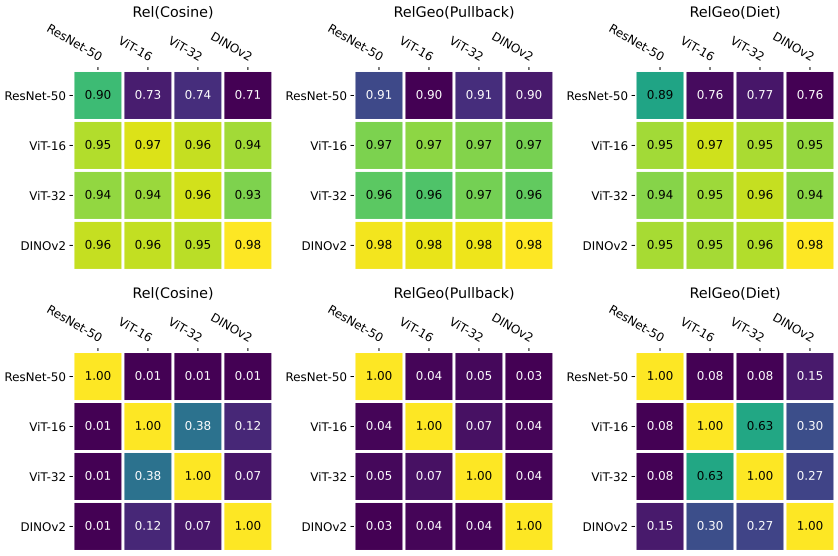


Figure 8: CIFAR-10 Accuracies, symmetricized MRR cdist

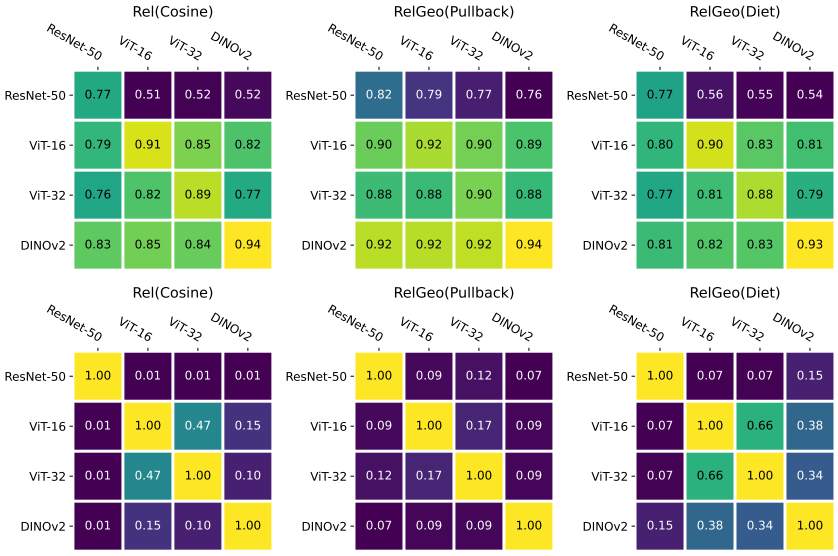


Figure 9: CIFAR-100 Accuracies, symmetricized MRR cdist

Table 8: MRR based on cdist.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
`Rel(Cosine)` Moschella et al. [2023]	0.051 ± 0.072	0.071 ± 0.107	0.078 ± 0.105	0.023 ± 0.02	0.02 ± 0.032
`RelGeo(Pullback)`	0.019 ± 0.005	0.04 ± 0.015	0.106 ± 0.11	0.108 ± 0.092	0.012 ± 0.005
`RelGeo(Diet)`	$\mathbf{0.127} \pm 0.118$	$\mathbf{0.151} \pm 0.138$	$\mathbf{0.298} \pm 0.141$	$\mathbf{0.269} \pm 0.195$	$\mathbf{0.123} \pm 0.103$

Table 9: Symmetrized MRRs based on cdist.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
`Rel(Cosine)` Moschella et al. [2023]	0.098 ± 0.133	0.122 ± 0.164	0.103 ± 0.146	0.046 ± 0.055	0.046 ± 0.081
`RelGeo(Pullback)`	0.046 ± 0.013	0.105 ± 0.031	0.179 ± 0.173	0.187 ± 0.141	0.04 ± 0.021
`RelGeo(Diet)`	$\mathbf{0.252} \pm 0.189$	$\mathbf{0.278} \pm 0.211$	$\mathbf{0.462} \pm 0.148$	$\mathbf{0.433} \pm 0.212$	$\mathbf{0.306} \pm 0.188$

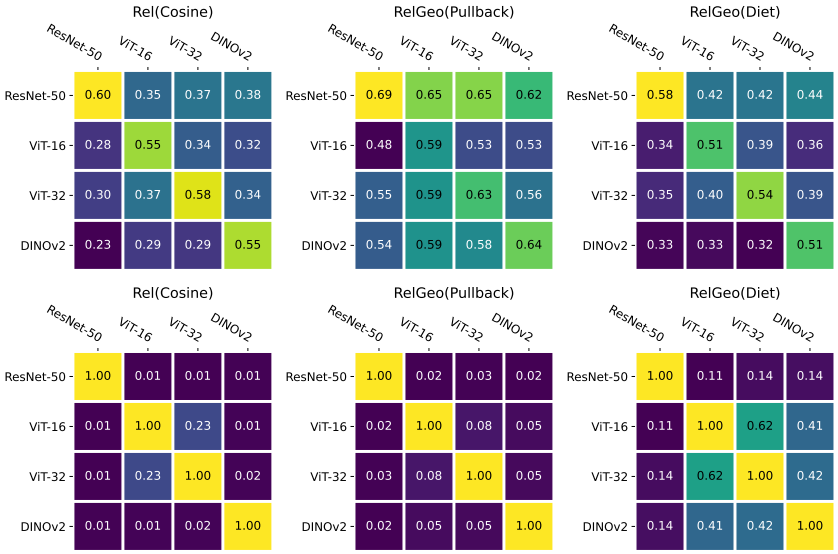


Figure 10: SVHN Accuracies, symmetricized MRR cdist

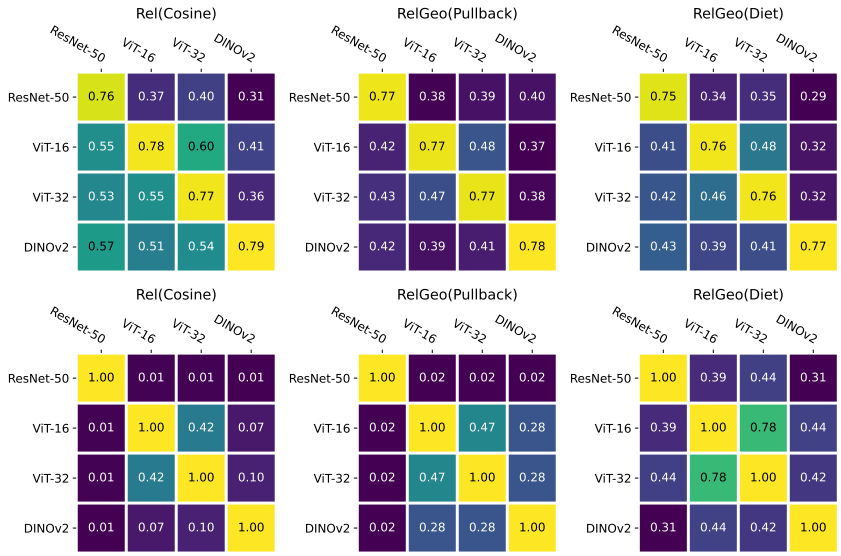


Figure 11: ImageNet Accuracies, symmetricized MRR cdist

Table 10: Accuracies as aggregated alternatively.

Method	ResNet-50	ViT-16	ViT-32	DINOv2
`Rel(Cosine)` Moschella et al. [2023]	0.507 ± 0.2	0.669 ± 0.229	0.664 ± 0.218	0.678 ± 0.24
`RelGeo(Pullback)`	$\mathbf{0.646} \pm 0.209$	$\mathbf{0.709} \pm 0.208$	$\mathbf{0.72} \pm 0.194$	$\mathbf{0.737} \pm 0.209$
`RelGeo(Diet)`	0.529 ± 0.194	0.658 ± 0.229	0.661 ± 0.219	0.668 ± 0.237

Table 11: MRRs based on cosine as aggregated alternatively.

Method	ResNet-50	ViT-16	ViT-32	DINOv2
`Rel(Cosine)` Moschella et al. [2023]	0.011 ± 0.005	0.138 ± 0.11	0.133 ± 0.112	0.147 ± 0.128
`RelGeo(Pullback)`	0.074 ± 0.077	0.107 ± 0.118	0.116 ± 0.126	0.079 ± 0.074
`RelGeo(Diet)`	$\mathbf{0.182} \pm 0.107$	$\mathbf{0.299} \pm 0.184$	$\mathbf{0.316} \pm 0.182$	$\mathbf{0.201} \pm 0.076$

A.2.3 Alternative aggregation

Here we consider an alternative way to aggregate the results, i.e. grouping by the models. The results are reported in Table 10, Table 12, Table 11, Table 14 and Table 13. In general, the observation remains: RelGeo(Pullback) yields good accuracies and RelGeo(Diet) yields good MRRs.

A.2.4 Number of anchors

We investigate the impact of the number of anchors. The results are shown in Figure 12 and Figure 13. The general conclusion that RelGeo(Pullback) is good in terms of accuracies, RelGeo(Diet) is good in terms of MRRs persist with varying number of anchors.

A.2.5 Number of diet points

We analyze the impact of the number of diet points. The results are shown in Figure 14. The performances of RelGeo(Diet) improve as the number of diet points become larger.

A.2.6 Number of discretization steps

We analyze the impact of the number of discretization steps on RelGeo(Pullback) and RelGeo(Diet) and provide the results in Figure 15 and Figure 16. The performances do not vary much depending on the discretization steps, though using multiple steps seem to help.

A.2.7 Diet augmentation strengths

We analyze the impact of different data augmentation strengths on RelGeo(Diet). The results are shown in Figure 17. Similar to the observations in terms of self-supervised learning [Ibrahim et al., 2024], RelGeo(Diet) benefits from stronger data augmentations.

Table 12: MRRs based on cdist as aggregated alternatively.

Method	ResNet-50	ViT-16	ViT-32	DINOv2
`Rel(Cosine)` Moschella et al. [2023]	0.007 ± 0.001	0.079 ± 0.086	0.089 ± 0.112	0.019 ± 0.015
`RelGeo(Pullback)`	0.023 ± 0.016	0.075 ± 0.096	0.078 ± 0.099	0.051 ± 0.05
`RelGeo(Diet)`	$\mathbf{0.121} \pm 0.094$	$\mathbf{0.253} \pm 0.193$	$\mathbf{0.258} \pm 0.194$	$\mathbf{0.143} \pm 0.065$

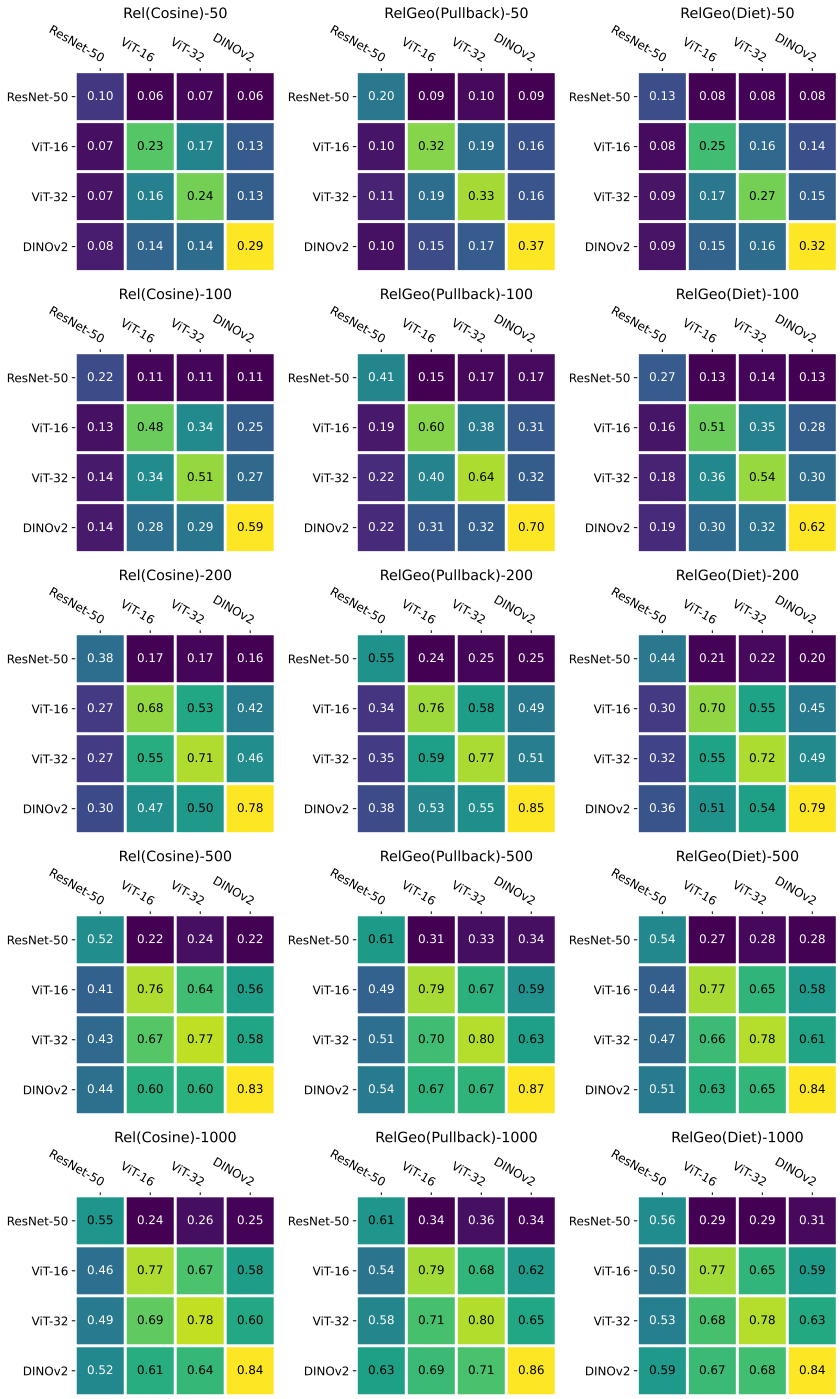


Figure 12: Accuracies on CUB with varying number of anchors. From top to bottom: 50, 100, 200, 500, 1000 anchors.

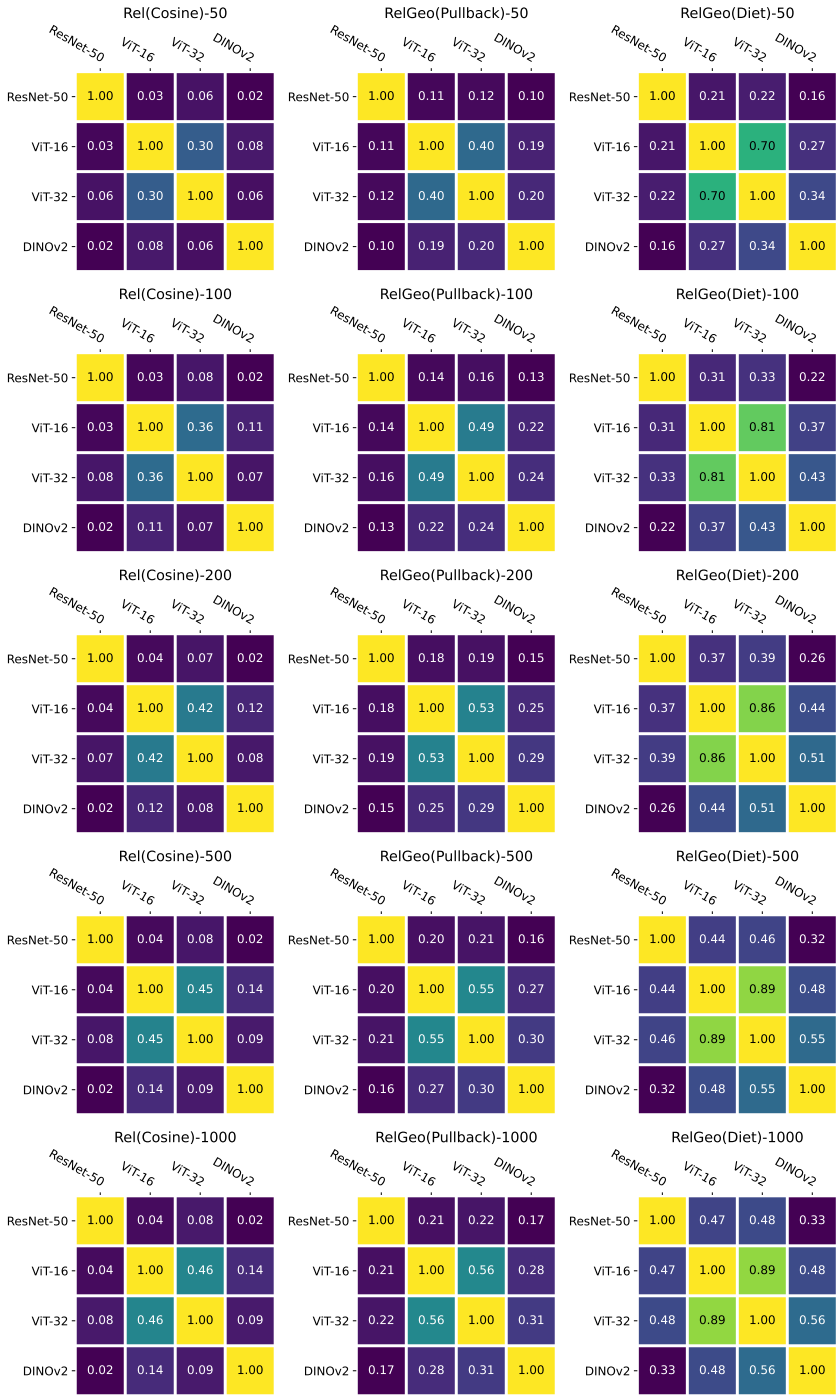


Figure 13: Symmetric MRR cosine on CUB with varying number of anchors. From top to bottom: 50, 100, 200, 500, 1000 anchors.

Table 13: Symmetric MRRs based on cosine as aggregated alternatively.

Method	ResNet-50	ViT-16	ViT-32	DINOv2
`Rel(Cosine)` Moschella et al. [2023]	0.032 ± 0.023	0.212 ± 0.173	0.208 ± 0.175	0.124 ± 0.104
`RelGeo(Pullback)`	0.143 ± 0.132	0.197 ± 0.186	0.205 ± 0.189	0.154 ± 0.137
`RelGeo(Diet)`	$\mathbf{0.336} \pm 0.143$	$\mathbf{0.506} \pm 0.2$	$\mathbf{0.526} \pm 0.187$	$\mathbf{0.42} \pm 0.106$

Table 14: Symmetric MRRs based on cdist as aggregated alternatively.

Method	ResNet-50	ViT-16	ViT-32	DINOv2
`Rel(Cosine)` Moschella et al. [2023]	0.009 ± 0.005	0.141 ± 0.156	0.134 ± 0.158	0.049 ± 0.047
`RelGeo(Pullback)`	0.052 ± 0.033	0.144 ± 0.146	0.147 ± 0.146	0.103 ± 0.085
`RelGeo(Diet)`	$\mathbf{0.204} \pm 0.13$	$\mathbf{0.432} \pm 0.235$	$\mathbf{0.437} \pm 0.232$	$\mathbf{0.313} \pm 0.108$

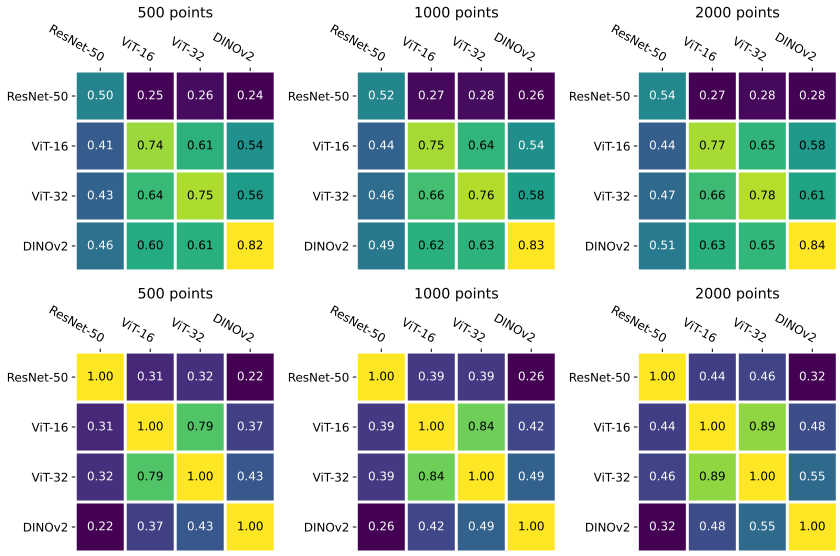


Figure 14: Results of RelGeo(Diet) on CUB with varying number of diet points. Top: accuracies; bottom: symmetric MRR cosine.

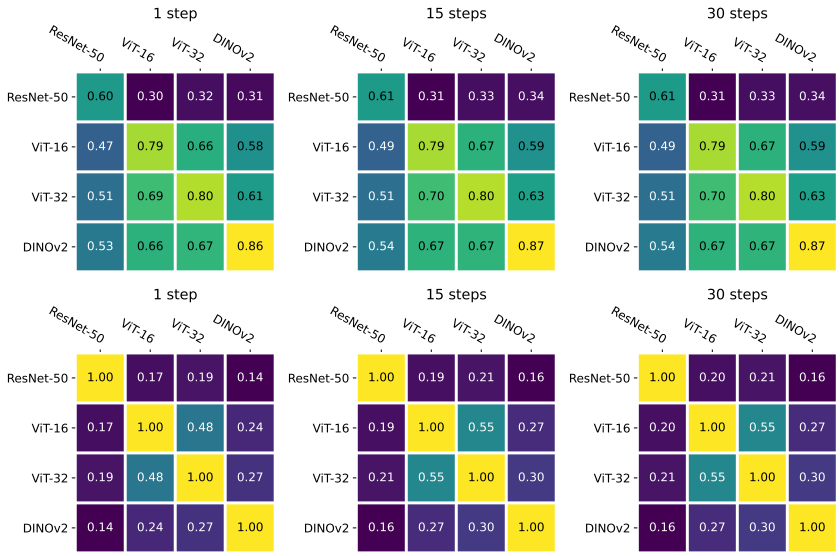


Figure 15: Results of RelGeo(Pullback) on CUB with varying number of discretization steps. Top: accuracies; bottom: symmetric MRR cosine.

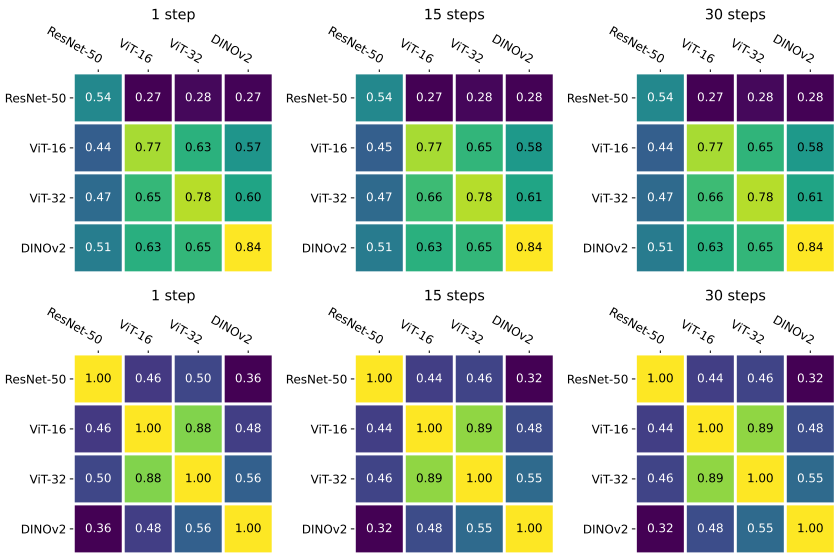


Figure 16: Results of RelGeo(Diet) on CUB with varying number of discretization steps. Top: accuracies; bottom: symmetric MRR cosine.

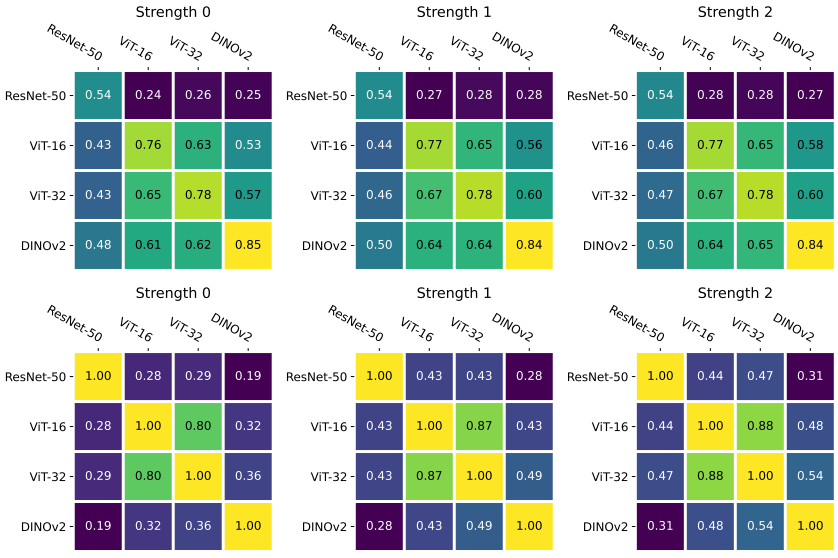


Figure 17: Results of RelGeo(Diet) on CUB with varying diet augmentation strengths. Results with strength 3 can be seen above. Top: accuracies; bottom: symmetric MRR cosine.