

Connecting Neural Models Latent Geometries with Relative Geodesic Representations

Hanlin Yu¹
University of Helsinki

Berfin Inal
University of Amsterdam

Georgios Arvanitidis
DTU

Søren Hauberg
DTU

Francesco Locatello
IST Austria

Marco Fumero¹
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, pretraining schemes and modalities. Code is available at <https://github.com/marco0git/RelativeGeodesics>.

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 given similar inputs [Li et al., 2015, Moschella et al., 2023, Fumero et al., 2024, Kornblith et al., 2019], 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, Löhner and Moeller, 2024, Moayeri et al., 2023]. This suggests a certain consistency in how neural nets encode information, emphasizing the importance of studying the

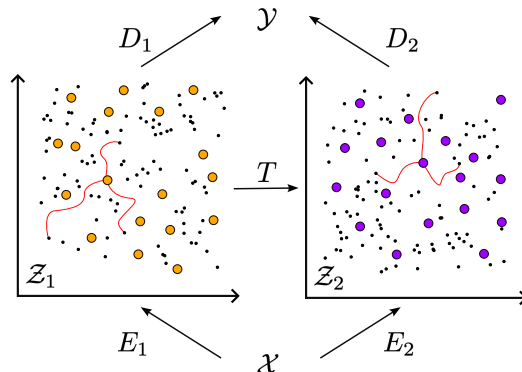


Figure 1: *Neural models trained on similar data learn parametrizations of the same manifold.* NNs learn parametrizations (D_1, D_2) of the same underlying manifold $\mathcal{Y}$ up to isometries T . Pulling back the metric from $\mathcal{Y}$ makes relative geodesic representations invariant to transformations T between latent spaces Z_1 and Z_2 .

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

internal representations and the transformations that relate them, to the extent to hypothesize whether neural nets are converging toward a unique 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 is 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, 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 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 show 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 vision foundation models, across different models, training schemes, and modalities, outperforming prior 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., 2019, 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, Csiszárík 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, Löhner and Moeller, 2024].

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 *pullback 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 Allasonnière, 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 the output space and pulling back the L_2 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 $\mathbf{z} = E_{\theta_1}(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$ is mapped from the input domain $\mathcal{X}$ to 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 $(\mathcal{X}, \mathcal{X}')$, a pair of neural models trained on them $(F_{\mathcal{X}}^1, F_{\mathcal{X}'}^2)$ and a partial correspondence between the domains $\Gamma : \mathcal{A}_{\mathcal{X}} \mapsto \mathcal{A}_{\mathcal{X}'}$ where $\mathcal{A}_{\mathcal{X}} \subset \mathcal{X}$ and $\mathcal{A}_{\mathcal{X}'} \subset \mathcal{X}'$, the problem of *latent space communication* is the one of finding a full correspondence $\Lambda : E^1(\mathcal{X}) \mapsto E^2(\mathcal{X}')$ between the two domains, from Γ . In a simplified setting, for example two models trained with different initialization or architectures on the same data, $\mathcal{X} = \mathcal{X}'$ and the correspondence is the identity. When $\mathcal{X} \neq \mathcal{X}'$ the problem recovers the multimodal setting.

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 $E_{\mathcal{X}} : \mathcal{X} \rightarrow \mathcal{Z}$, a set of anchors $\mathcal{A}_{\mathcal{X}} \subset \mathcal{X}$, and a similarity or distance function $d : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}$, the *relative representation* for a sample $\mathbf{x} \in \mathcal{X}$ is:

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

where $\mathbf{z} = E_{\mathcal{X}}(\mathbf{x})$, and $\bigoplus$ denotes row-wise concatenation. In the original method [Moschella et al., 2023], d corresponds to 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 Ch.2.4 of Do Carmo and Flaherty Francis [1992]).

Formally, the decoder $D : \mathcal{Z} \mapsto \mathcal{Y}$ takes as input a latent representation $\mathbf{z} \in \mathcal{Z}$ and outputs $\mathbf{y}$. Given a Riemannian metric defined on $\mathcal{Y}$ as $G_{\mathcal{Y}}(\mathbf{y})$, one can obtain the Riemannian metric at $\mathbf{z}$ as:

$$G_{\mathcal{Z}}(\mathbf{z}) = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{z}} \right)^{\top} G_{\mathcal{Y}}(\mathbf{y}) \left(\frac{\partial \mathbf{y}}{\partial \mathbf{z}} \right) = J_D(\mathbf{z})^{\top} G_{\mathcal{Y}}(\mathbf{y}) J_D(\mathbf{z}),$$

where $J_D(\mathbf{z})$ is the Jacobian of D at $\mathbf{z}$. The metric tensor $G_{\mathcal{Y}}$ 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 defined as:

$$L(\gamma) = \int_a^b \sqrt{\mathbf{v}(t)^{\top} G_{\mathcal{Y}}(\gamma(t)) \mathbf{v}(t)} dt, \quad (1)$$

where $\mathbf{v}(t) = \dot{\gamma}(t)$. A slight variation of the above functional gives the geodesic energy $\mathcal{E}$ of γ [Arvanitidis et al., 2018, Shao et al., 2018]

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_a^b \mathbf{v}(t)^{\top} G_{\mathcal{Y}}(\gamma(t)) \mathbf{v}(t) dt. \quad (2)$$

Both can be discretized and approximated in practice using finite difference approaches [Yang et al., 2018, Shao et al., 2018]. Geodesics minimize both the length and the energy, where for optimization the latter is usually preferred for numerical stability [Haugberg, 2025]. These quantities have the property of being *invariant* to certain reparametrizations, as formalized in the following proposition:

Proposition 3.1. Let $\gamma : [0, 1] \rightarrow \mathcal{M}$ be a smooth curve on a Riemannian manifold $(\mathcal{M}, G)$, and let $(\mathcal{M}', G')$ be a reparameterization of the manifold and $\varphi : [0, 1] \rightarrow [0, 1]$ a smooth, strictly increasing reparametrization of γ . Setting $\gamma'(\tau) = \gamma(\varphi(\tau))$ the Riemannian length and energy of γ are invariant under reparameterizations of the manifold:

$$\begin{aligned}\mathcal{E}[\gamma] &= \frac{1}{2} \int_0^1 \left\| \frac{d\gamma}{d\tau} \right\|_G^2 d\tau = \frac{1}{2} \int_0^1 \left\| \frac{d\gamma'}{d\tau} \right\|_{G'}^2 dt, \\ L[\gamma] &= \int_0^1 \left\| \frac{d\gamma}{d\tau} \right\|_G d\tau = \int_0^1 \left\| \frac{d\gamma'}{d\tau} \right\|_{G'} dt.\end{aligned}$$

Furthermore, the Riemannian arc length of γ is invariant under reparametrizations γ' on $\mathcal{M}$:

$$L[\gamma'] = \int_0^1 \left\| \frac{d\gamma'}{d\tau} \right\|_G d\tau = \int_0^1 \left\| \frac{d\gamma}{dt} \right\|_G dt = L[\gamma].$$

We provide the proof in Appendix A.1.1.

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 , distance $d_{\mathcal{X}}$ induced by metric $G_{\mathcal{Y}}$, steps N , step size Δt , mode $\in \{\text{energy}, \text{distance}\}$
Ensure: $RR^{geo}(x; \mathcal{A}_{\mathcal{X}})$
1: $z \leftarrow E(x)$, $RR^{geo} \leftarrow []$
2: **for** $a \in \mathcal{A}_{\mathcal{X}}$ **do**
3: $z_a \leftarrow E(a)$, $d \leftarrow 0$
4: **for** $j = 1$ to N **do**
5: $\gamma_j \leftarrow (1 - \frac{j}{N})z + \frac{j}{N}z_a$
6: $\gamma_{j-1} \leftarrow (1 - \frac{j-1}{N})z + \frac{j-1}{N}z_a$
7: $v \leftarrow D(\gamma_j) - D(\gamma_{j-1})$
8: $G \leftarrow G_{\mathcal{Y}}(D(\gamma_j))$
9: $s \leftarrow v^{\top} G v$
10: $d \leftarrow d + \Delta t \cdot (\text{energy} \Rightarrow \frac{1}{2}s, \text{distance} \Rightarrow \sqrt{s})$
11: **end for**
12: Append d to RR^{geo}
13: **end for**
14: **return** RR^{geo}

From 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 $F_{\mathcal{X}}^1, F_{\mathcal{X}'}^2$. The relative representation framework captures this transformation implicitly if equipped with the right metric: we posit that 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 T of the manifolds $\mathcal{M}_1, \mathcal{M}_2$. However, for high-dimensional problems, the high cost of computing the geodesic (corresponding to minimizing Eq. 2) makes this impractical [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, for example the geodesics going outside of the data manifold.

We therefore approximate the geodesic quantities by directly considering the energy (or the length) of the straight line (in the Euclidean sense) connecting representations in the latent space:

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

where $\tilde{\gamma}_{\alpha}(z_1, z_2) = (1 - \alpha)z_1 + \alpha z_2$ is the convex combination between the points z_1, z_2 . The approximation gives a natural upper bound to the geodesic distance: for $\tilde{\gamma}$ it can be shown to relate to the arc length of a curve defined in Eq. 1 and the energy in Eq. 2 using the following bounds:

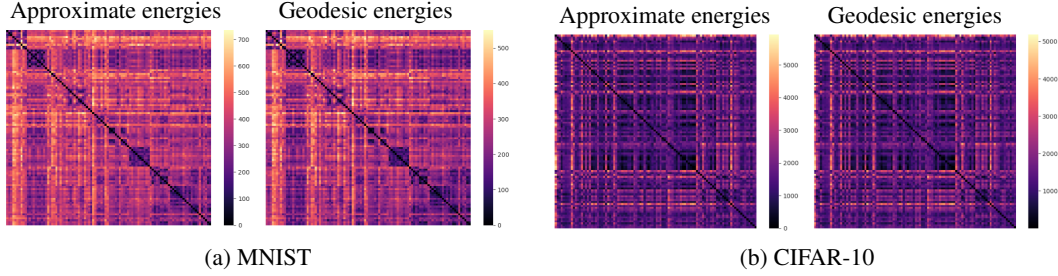


Figure 2: Pairwise latent-space energy matrices for (a) MNIST and (b) CIFAR-10. In each subfigure, the left heatmap shows the straight-line energy approximation and the right shows the geodesic energies of the ground truth geodesic curve. The Spearman rank correlations between the two measures are $\rho = 0.99$ for MNIST and $\rho = 1.00$ for CIFAR-10, demonstrating near-perfect agreements.

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

The proof is in Appendix A.1.2. Moreover the approximation is far more *efficient* to compute, without requiring minimization of equation 2, and is *accurate*, as empirically verified in Figure 2.

Discretization. When the step size is small enough, energy and arc length in the latent space as in Equations 1,2 can be approximated by their counterpart on the output space using discretized finite difference schemes [Shao et al., 2018]:

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

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

where $\Delta t = \frac{1}{N}$, with N being the number of discretization steps. For Euclidean geometry, the geodesic arc lengths are given in closed form as the geodesics are straight lines. Unlike the energy, the curve length is invariant under reparametrizations (proposition 3.1). As such we focus on the curve length in our experiments. The resulting algorithm is summarized in Algorithm 1. In practice, with specific choice of $G_{\mathcal{Y}}$ one can avoid approximating the distance between $D(\gamma_j, \gamma_{j-1})$ explicitly using $G_{\mathcal{Y}}$ by directly calculating the distance or energy between γ_j and γ_{j-1} on $\mathcal{Y}$.

Approximate geodesic energies. Our choice comes with three advantages: (i) *efficiency*: avoiding minimization of Eq.2 the computation for every sample reduces to a single forward pass for every discretization step γ and for each anchor, resulting in overall complexity of $O(TA)$ forward passes of the decoder, where A the number of anchors is the number of discretization steps. (ii) Directly using the arc length ensures *invariance* to reparametrizations of the manifold, matching our assumptions. (iii) As we only need reasonably accurate estimates of the arc lengths rather than the geodesic trajectory, the approach is *accurate*. Specifically, to assess how close the straight line energy approximation (2) is to the true geodesic energies, we first encoded 100 samples (10 per class, sorted by label) from MNIST [Deng, 2012] and CIFAR-10 using a simple convolutional autoencoder (architecture detailed in Appendix A.3.2). We then computed pairwise geodesic energy matrices over these latent representations using both methods, and the results are displayed in Fig. 2. Visually, both energy 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 A.3.5 for correlation results across different numbers of discretization steps and for implementation details).

3.3 Choice of pullback metric

The properties of the relative geodesic representations are determined by (i) the choice of the output space, (ii) the choice of the metric to pullback from the output space and (iii) the pretraining objective (e.g. reconstruction or classification) on which the decoder was trained.

Generative models. For models trained on a reconstruction loss such as autoencoders, or on generative objectives, such as variational autoencoders Kingma and Welling [2013], pulling back

metrics such as $L2$ distance have been shown to effectively reflect the underlying geometry of the latent space [Tosi et al., 2014, Arvanitidis et al., 2018, Hauberg, 2019].

Discriminative models. For discriminative models, such as classifiers or instance based discriminative models [Ibrahim et al., 2024], 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], possibly rendering the resulting geometry troublesome. We empirically inspect this approach in Appendix A.4.11. In this work we consider two principled approaches for discriminative models based on classification decoder heads and instance discrimination heads.

Pulling back from classifiers. Perhaps the most natural idea is, as discussed in Section 3.1, to construct a pullback metric based on the model’s outputs, by simply pulling back the euclidean $L2$ metric from the logit space of the classifier. Given an arbitrary encoder model, we train a classification head upon the latent representations (extracted e.g. from the last layer) and pulling back the euclidean $L2$ metric from the output logits of the head. The resulting relative geodesics representation will inherit properties of both the decoder head (up to class information) and the pretrained encoder.

Pulling back from instance discrimination decoders. Diet [Ibrahim et al., 2024] is a self-supervised training method which has been shown to learn representations with strong generalization to downstream tasks, and yield identifiability guarantees [Reizinger et al., 2025]. Specifically, in the infinite data limit representations from the Diet objective align the cluster centers of von-Mises Fisher (vMF) distributions, which lie on a unit sphere. The loss based on simple instance discrimination is:

$$\mathcal{L}_{Diet} = \mathbb{E}_{\mathbf{x}, i} \left[-\log \frac{\exp(\mathbf{w}_i^\top f(\mathbf{x}))}{\sum_j \exp(\mathbf{w}_j^\top f(\mathbf{x}))} \right], \quad (6)$$

where $\mathbf{W}$ is a linear projection and f is a nonlinear map. Furthermore, assigning the same instance label to data augmentations was shown beneficial to improve invariance. While it was originally proposed to train the entire neural network [Ibrahim et al., 2024], we instead use it to learn a decoder D on top of the pretrained neural network latent representations, by setting $D = f \circ \mathbf{W}$. To construct relative geodesic representations, we propose to pullback the spherical metric from the penultimate layer of the diet decoder, before the projection head $\mathbf{W}$. Further discussions on Diet can be found in Appendix A.2.1.

Both classifier and instance discriminator approaches discussed above use proper pullback metrics and fall under our proposed framework of relative geodesic representations: these representations inherit semantic information, up to class or instance level, from the decoder, while retaining structure of the pretrained encoder. Notably, when the encoder is pretrained, the relative geodesics representations are computationally efficient, since the decoder remains lightweight and the approximate geodesic energy computation is cheap. As we demonstrate in the following experimental sections, this approach yields meaningful and identifiable representations that are consistent across models.

4 Experiments

In the following sections we will evaluate the performance of relative geodesic representations on two instances of the latent communication problem, across models with different initializations, architectures, sizes and modalities.

Tasks description. We evaluate our approach on two representative instantiations of the latent communication problem: *retrieval* and *neural stitching*. In a retrieval setting we aim to solve the latent communication problem up to the *instance* level. Given pairs of model encoders $(E_{\mathcal{X}}^1, E_{\mathcal{X}'}^2)$ and access to their latent representations, we seek to recover a full correspondence Λ starting from a partial one Γ . For neural stitching the goal is to solve the latent communication problem up to the *task-label* level. Classical stitching approaches train an adapter Ψ between intermediate components of distinct neural networks so that $D_{\mathcal{X}'}^2 \circ \Psi \circ E_{\mathcal{X}}^1$ remains functional on a downstream task (e.g., classification). In Section 4.2, we operate in the *zero-shot stitching* regime [Moschella et al., 2023], where no adapter is trained explicitly. Instead, we solve implicitly for the transformations between representations by mapping them into *relative representation spaces*. This enables stitching pairs of models without any fine-tuning or additional supervision.

In Section 4.1, we evaluate *relative geodesic representations* on generative models, focusing on autoencoders. This analysis examines performance across networks trained with different initializations and datasets. In Section 4.2, we extend the evaluation to *discriminative foundation models*, assessing performance at scale across diverse architectures, pretraining objectives (e.g., self-supervised and classification), datasets, and modalities.

4.1 Experimental evaluation on autoencoders

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

4.1.1 Aligning independently trained neural representational spaces

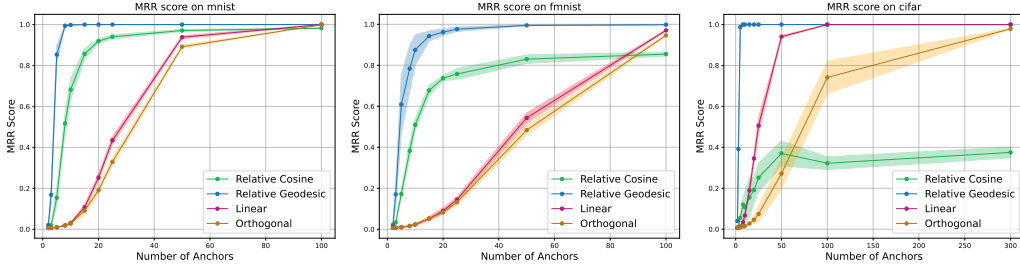


Figure 3: *Aligning latent spaces of autoencoders*: MRR score as a function of the number of anchors on pairs of autoencoders trained with different initializations on the MNIST (left), FashionMNIST (center), CIFAR10 (right) datasets, respectively. In green, we plot the performance of Moschella et al. [2023]; in red and orange the linear and orthogonal baselines respectively; in blue, our method. The shaded area indicates standard deviation across 5 different random sets of anchors. Relative geodesic consistently outperforms baselines, obtaining peak performance.

Setting. For the following experiment, we trained pairs of convolutional autoencoders (F_1, F_2) with different initializations on MNIST [Deng, 2012], FashionMNIST [Xiao et al., 2017], CIFAR10 [Krizhevsky, 2009] datasets. The architecture of the convolutional autoencoder is detailed in Appendix A.3.2. 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 $\Gamma : \mathcal{A}_X \mapsto \mathcal{A}_Y$, the objective is to evaluate how well it is possible to recover the full correspondence Λ 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], and with fitting a linear or orthogonal mapping using Γ .

Analysis of results. Fig. 3 plots the performance in terms of MRR on MNIST, FashionMNIST and 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.3.1) 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 Moschella et al. [2023], 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, sample efficiency and robustness.

4.1.2 Stitching autoencoder models

Setting. We consider the same pairs of autoencoders trained on the MNIST, FashionMNIST, CIFAR10 datasets of Section 4.1.1. Starting from a set of five random anchors, we estimate a transformation T between the model representational spaces $\mathbf{Z}_1, \mathbf{Z}_2$. In this experiment, to keep differently from Moschella et al. [2023], in which zero-shot stitching was achieved by training once a decoder

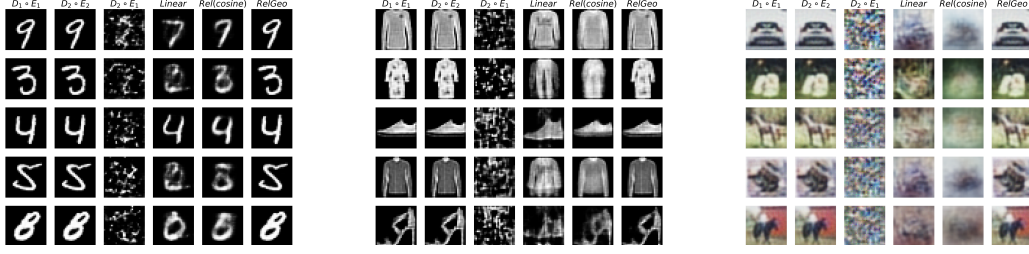


Figure 4: *Stitching on Autoencoders*: We visualize qualitative reconstructions of samples, stitching autoencoders of models trained with different initializations on MNIST (left), FashionMNIST (center), CIFAR10 (right). The first two columns show reconstructions from the original models; middle three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

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(L2)	0.047 ± 0.013	0.112 ± 0.031	0.412 ± 0.09	0.28 ± 0.129	0.025 ± 0.012
RelGeo(Diet)	0.387 ± 0.145	0.445 ± 0.142	0.566 ± 0.111	0.523 ± 0.177	0.314 ± 0.188

module with relative representations and then exchanging different encoder modules, here we achieve stitching without training any decoder. We compute relative representations 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 representation 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 against Moschella et al. [2023], Löhner and Moeller [2024], 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. As shown in Fig. 4, 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. Relative geodesics enable stitching of neural modules trained with different initializations.

4.2 Experiments on vision foundation models

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

4.2.1 Matching representational spaces of discriminative foundation models

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

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., 2016], 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 representations pulling back from Euclidean logits denoted as RelGeo(L2), and pulling back the spherical metric using a Diet decoder denoted as 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. While RelGeo(L2) may result in worse MRR numbers, RelGeo(Diet) provides consistently improved retrieval performance. In Appendix A.4.1 we report full results for the datasets.

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

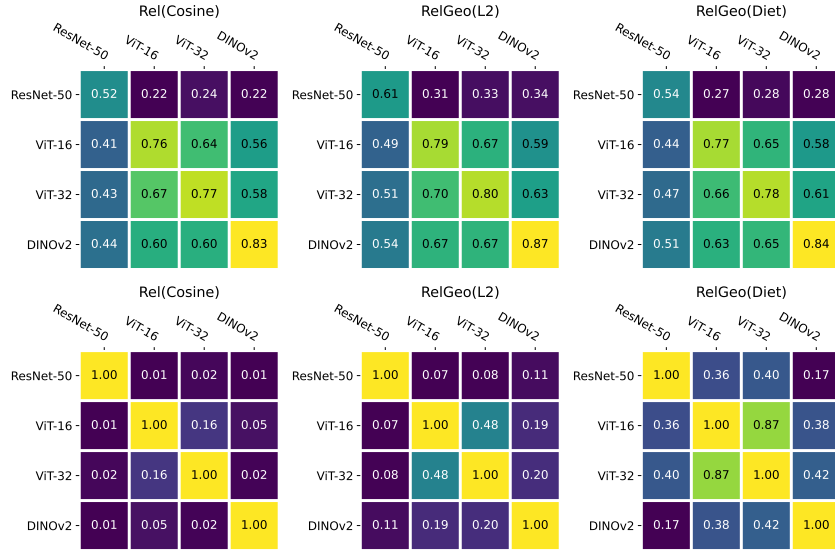


Figure 5: CUB Accuracies (top) and symmetricized MRR cosine (bottom). RelGeo(Diet) and especially RelGeo(L2) provide strong stitching accuracies, while RelGeo(Diet) maintains strong instance identifiability.

4.2.2 Zero-shot stitching of vision foundation models

Table 2: Average stitching performances across different settings. RelGeo(L2) 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	0.549 ± 0.152	0.531 ± 0.188	0.384 ± 0.115
RelGeo(L2)	0.955 ± 0.03	0.874 ± 0.055	0.501 ± 0.159	0.595 ± 0.163	0.59 ± 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 i th 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 the decoder D , e.g., a classification head.

Setting. We perform experiments on pretrained vision foundation models from Hugging Face Transformers [Wolf et al., 2020], investigating how well we can match representations together for

classification with different backbones with classification heads, on the same datasets and models as considered in Section 4.2.1, similarly comparing $\text{Rel}(\text{Cosine})$, $\text{RelGeo}(L2)$ and $\text{RelGeo}(\text{Diet})$.

Analysis of results. The results of the different methods across the different datasets 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 $\text{RelGeo}(L2)$ and $\text{RelGeo}(\text{Diet})$ provide strong stitching accuracies, with $\text{RelGeo}(L2)$ reflecting the benefits of pulling back class specific information. $\text{RelGeo}(\text{Diet})$ still results in good accuracies while having very strong MRR metrics, as shown in 1.

Takeaway. Relative geodesic representations yield good accuracies and good MRRs, avoiding downgrading of performance when performing model stitching while retaining sample identifiability.

4.2.3 Matching different modalities

In this section we evaluate relative geodesic representations in the multimodal setting.

Setting. We study the retrieval task in terms of vision foundation models and the text encoders of CLIP [Radford et al., 2021] with both patch 16 and patch 32, using Flickr30k dataset [Young et al., 2014]. Keeping the text encoder of CLIP fixed, we swap the vision encoder with the ones of ResNet-50, DINOv2 and ViT, including different patch and model sizes. Due to the lack of class labels, $\text{RelGeo}(L2)$ is not applicable, and we compare $\text{RelGeo}(\text{Diet})$ with $\text{Rel}(\text{Cosine})$. While we observed that using data augmentations is beneficial for $\text{RelGeo}(\text{Diet})$, due to the lack of a principled approach to construct data augmentations on texts corresponding to image augmentations, we do not employ augmentations.

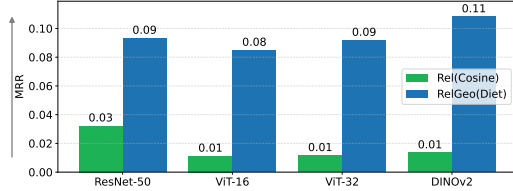


Figure 6: *Matching multimodal models.* Symmetricized MRR cosine on Flickr30k. $\text{RelGeo}(\text{Diet})$ substantially improves upon $\text{Rel}(\text{Cosine})$ in aligning multimodal models.

Analysis of results. The results in terms of symmetricized MRR metric are shown in Figure 6. We observe that $\text{RelGeo}(\text{Diet})$ yields significantly improved stitching performances upon $\text{Rel}(\text{Cosine})$. In Appendix A.4.9 we show the full pairwise matrices of MRR, comprising of the unimodal performances, inter vision models, and text models.

Takeaway. Relative geodesic representations show promising results for obtaining identifiable representations in multimodal scenarios.

5 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 approximating geodesics using straight-line arc length (or energy) can deteriorate in regions of high curvature in the latent space, typically corresponding to areas far from the support of the training data. Moreover, this could require increasingly smaller step sizes, hurting the efficiency performance of the method. This suggests exploring nonlinear paths, 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 geodesic representation has the interesting property of restricting the alignment problem to the information relevant to the decoding task. This could be useful to (i) further explore no training 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 Krzmar, 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>.
- B. C. Brown, A. L. Caterini, B. L. Ross, J. C. Cresswell, and G. Loaiza-Ganem. Verifying the Union of Manifolds Hypothesis for Image Data. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=Rvee9CAX4fi>.
- 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. Allasonnière. 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. Csizsárik, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- M. Huh, B. Cheung, T. Wang, and P. Isola. The platonic representation hypothesis, 2024. URL <https://arxiv.org/abs/2405.07987>.

- A. Hyvärinen, I. Khemakhem, and H. Morioka. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns*, 4(10):100844, 2023. ISSN 2666-3899.
- 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>.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- 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, 2019.
- 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, 2024. 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. Castriato, 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.
- B. M. G. Nielsen, E. Marconato, A. Dittadi, and L. Gresele. When Does Closeness in Distribution Imply Representational Similarity? An Identifiability Perspective, 2025. URL <https://arxiv.org/abs/2506.03784>. _eprint: 2506.03784.
- 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>.
- R. Pascanu, C. Lyle, I.-V. Modoranu, N. E. Borrás, D. Alistarh, P. Velickovic, S. Chandar, S. De, and J. Martens. Optimizers Qualitatively Alter Solutions And We Should Leverage This, 2025. URL <https://arxiv.org/abs/2507.12224>. _eprint: 2507.12224.
- A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- P. Reizinger, A. Bizeul, A. Juhos, J. E. Vogt, R. Balestrieri, 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=hrqN0xpItr>.
- G. Roeder, L. Metz, and D. Kingma. On Linear Identifiability of Learned Representations. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9030–9039. PMLR, July 2021.
- 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.
- P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tac1_a_00166. URL <https://aclanthology.org/Q14-1006/>. Place: Cambridge, MA Publisher: MIT Press.
- 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 Proof of theoretical results

A.1.1 Proof of Proposition 3.1

Proof. We first prove the first half, i.e. the invariance of Riemannian curve length and energy across reparameterizations of the manifold. This can be proven by observing that the inner product at a point along the curve is invariant across such reparameterizations:

$$\begin{aligned}\|\dot{\mathbf{x}}\|_G &= \dot{\mathbf{x}}^\top G(\mathbf{x}) \dot{\mathbf{x}} = \left(\frac{d\mathbf{x}}{d\mathbf{x}'} \dot{\mathbf{x}}'\right)^\top \left(\frac{d\mathbf{x}'}{d\mathbf{x}}\right)^\top G'(\mathbf{x}') \frac{d\mathbf{x}'}{d\mathbf{x}} \frac{d\mathbf{x}}{d\mathbf{x}'} \dot{\mathbf{x}}' \\ &= \dot{\mathbf{x}}'^\top G'(\mathbf{x}') \dot{\mathbf{x}}' = \|\dot{\mathbf{x}}'\|_{G'}.\end{aligned}$$

As such, the length and the energy of the same curves on different manifolds are integrals of the same quantities, hence are equal.

We then prove the second half, i.e. the invariance of Riemannian curve length across reparameterizations of the curve. Based on Equation 4.7 from Hauberg [2025], we have

$$\begin{aligned}L[\gamma'] &= \int_0^1 \left\| \frac{d\gamma}{d\tau} \right\|_G d\tau = \int_0^1 \left\| \frac{d\gamma}{d\tau} \right\|_G \frac{\varphi(t)}{t} dt \\ &= \int_0^1 \left\| \frac{d\gamma}{d\tau} \frac{\varphi(t)}{t} \right\|_G dt = \int_0^1 \left\| \frac{d\gamma}{dt} \right\|_G dt \\ &= L[\gamma].\end{aligned}$$

□

A.1.2 Proof of Equation 3

Proof. We first prove $d(\mathbf{z}_0, \mathbf{z}_1)^2 \leq L^2(\tilde{\gamma})$, then prove $L^2(\tilde{\gamma}) \leq 2\mathcal{E}(\tilde{\gamma})$.

For the first part, according to the definition of geodesic distance, we have $d(\mathbf{z}_0, \mathbf{z}_1) \leq L(\tilde{\gamma})$ and, as such, $d(\mathbf{z}_0, \mathbf{z}_1)^2 \leq L^2(\tilde{\gamma})$.

The second part involves $L^2(\tilde{\gamma}) \leq 2\mathcal{E}(\tilde{\gamma})$, which can be proven using the Cauchy-Schwarz inequality. See Equation 7.14 in [Hauberg, 2025], where we denote $\mathbf{u}_t = \frac{d\tilde{\gamma}}{dt}$ and $\mathbf{v}_t = 1$.

$$\begin{aligned}L^2(\tilde{\gamma}) &= \int_0^1 \left\| \frac{d\tilde{\gamma}}{dt} \right\|^2 dt = \langle \mathbf{u}, \mathbf{v} \rangle \leq \|\mathbf{u}\| \|\mathbf{v}\| = \sqrt{\int_0^1 \left\| \frac{d\tilde{\gamma}}{dt} \right\|^2 dt} \sqrt{\int_0^1 1^2 dt} = \sqrt{\int_0^1 \left\| \frac{d\tilde{\gamma}}{dt} \right\|^2 dt} \\ &= 2\mathcal{E}(\tilde{\gamma}).\end{aligned}$$

□

A.2 Additional explanations

A.2.1 Details on Diet

Proposed as a self-supervised learning method [Ibrahim et al., 2024], Diet was also shown to yield interesting identifiable guarantees [Reizinger et al., 2025], laying the theoretical foundation for RelGeo(Diet), where we employ the resulting geometry.

One can consider such a scenario [Reizinger et al., 2025]: some latent variables $\mathbf{z}$ are drawn from a vMF distribution, and pushed forward through a continuous and injective generator function g to obtain the data $\mathbf{x}$. Remarkably, given only $\mathbf{x}$ without the knowledge of g , it is possible to (to some degree) recover the latent variables $\mathbf{z}$ through parameterizing a model and optimizing the instance discrimination loss as given in Equation 6. Specifically, suppose there is a finite set of vectors $\mathbf{v}_c$ on a unit sphere, each representing a class, and a finite set of instances. One instance belongs to

exactly one class, and every class is employed by some instance. Additionally, the instance labels are chosen uniformly, and the latent variables $\mathbf{z}$ are drawn from a vMF distribution centered around the corresponding cluster vector $\mathbf{v}_c$ with concentration parameter κ .

Then, after the model is trained using the loss function as in Equation 6, when both f and w are not unit-normalized, $f \circ g$ is linear. This can be proven rigorously by expanding upon the theoretical framework of non-linear ICA [Hyvärinen et al., 2023]. As such, we propose to utilize $f \circ g$ to form the representations. For further technical details on the assumptions and additional results, we refer interested readers to Reizinger et al. [2025].

Assuming spherical geometry, the distance between two points $\mathbf{x}$ and $\mathbf{y}$ can be computed as

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

In the above formula, points that do not precisely lie on the unit sphere are effectively projected onto it. Interestingly this bears a strong resemblance to the cosine distances as used in the original paper on relative representations [Moschella et al., 2023].

A.2.2 Why it works

Prior work on representational alignment has shown that representations from different models can often be approximately aligned using simple transformations, e.g. linear, orthogonal or locally linear maps. Even when models are trained independently – with different architectures, modalities, or datasets that nonetheless share an underlying structure – they tend to learn similar representations, suggesting convergence towards a shared encoding of entities [Huh et al., 2024]. From a theoretical standpoint, identifiability results [Roeder et al., 2021] imply that if two discriminative models learn the same likelihood function, their internal representations must be equivalent up to a linear transformation. However, this ideal scenario rarely holds exactly in practice: training dynamics, nuisance factors, and unmodeled variability can all introduce distortion. In our case, it may be too strong to assume that two models learn different parameterizations of an identical manifold. Instead, we adopt a weaker assumption, that they do so up to some bounded distortion. Recent theoretical work has begun to explore relaxations of strict identifiability to account for such bounded distortions [Nielsen et al., 2025]. Integrating these relaxations into our Riemannian framework presents a promising direction for future work.

In general, the few theoretical results available, e.g. [Roeder et al., 2021], often rely on unrealistic assumptions, e.g. proofs in axiomatic settings, infinite-data regimes, or the requirement that two models learn exactly the same likelihood function. In our view, a meaningful first step toward bringing theory and practice is to relax these assumptions, as initiated in [Nielsen et al., 2025], and begin to model more realistic scenarios, e.g. including the dynamics introduced by model training.

We believe that Riemannian methods can play a key role in this direction to try to capture local alignments beyond linear global transformations of the space as considered in Roeder et al. [2021] and possibly accounting for distortions measured in the linear space in practice. Nevertheless, we remark that neural networks could find qualitatively different solutions [Pascanu et al., 2025] and that the union of manifolds hypothesis might be more appropriate for modeling image data [Brown et al., 2023].

A.3 Additional details

A.3.1 Mean Reciprocal Rank

Mean Reciprocal Rank (MRR) is a commonly used metric to evaluate the performance of retrieval systems, and has been used to evaluate the capabilities of representations for instance discrimination [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:

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

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 $D = \frac{1}{2} (D^\top + D)$. In Section 4.2.1 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.3.2 Architectural details

We provide in Table 3 the architectural details of the convolutional autoencoders employed in experiments in Figures 3 and 4.

Table 3: Architecture of the convolutional autoencoders.

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

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.

Table 4: Architecture of the simple decoders.

Classification head
$input_dim$ LayerNorm
$input_dim \times 500$ Linear-Tanh
$500 \times num_classes$ Linear

Table 5: Architecture of the decoders for evaluations.

Final classification head
$input_dim$ LayerNorm
$input_dim \times input_dim$ Linear-Tanh
InstanceNorm1d
$input_dim \times num_classes$ Linear

A.3.3 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.3.4 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 were conducted on a single NVIDIA RTX 3080TI GPU. Experiments involving vision foundation models were run on a compute cluster, each job using a single NVIDIA A100 GPU and 10 CPU cores, with runtimes of several hours. Preliminary experiments required additional resources, and in total we estimate having used several hundred GPU hours.

Further ablation studies on the running times can be found in Section A.4.10.

A.3.5 Geodesic approximation

Here, we provide the experimental details of the results presented in Fig. 2 and Fig. 7. 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 fixed a random seed of 42 to ensure reproducibility.

Energy 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 has Apache-2.0 license, which wraps the decoder into a pullback manifold, initializes a parameterized spline path between codes, and then optimizes its parameters to minimize the Riemannian energy. Geodesic energies are computed as in Eq. 2. Pairwise energies are computed and visualized in Figures 2 and 7, 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. 7, 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

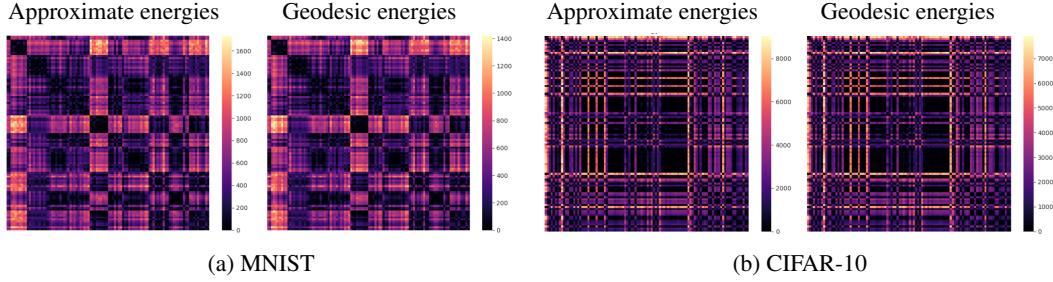


Figure 7: Pairwise latent-space energy matrices for (a) MNIST and (b) CIFAR-10, with latent dimensionality 2. In each subfigure, the left heatmap shows the straight-line energy proxy and the right shows the full Riemannian geodesic energies. The Spearman rank correlation between the two measures is 0.99 for MNIST and $\rho = 1.00$ for CIFAR-10, demonstrating near-perfect agreement.

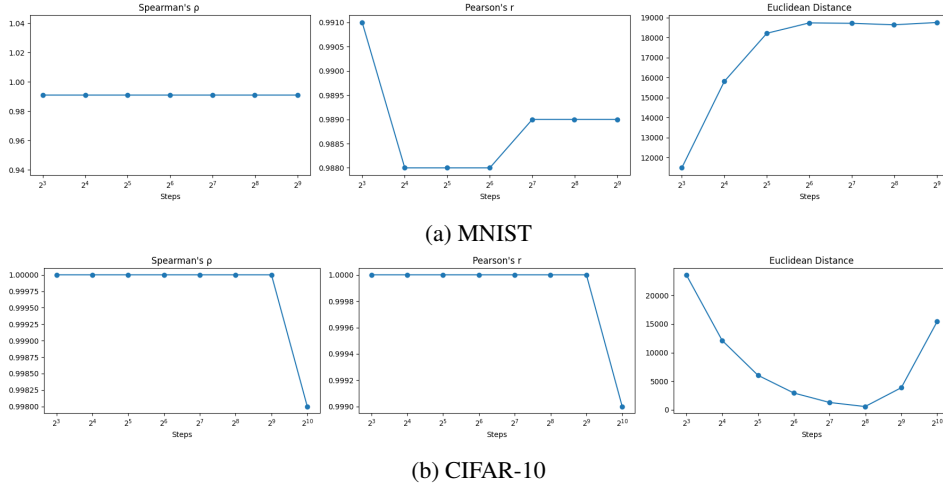


Figure 8: Impact of varying discretization levels on similarity and energy metrics for (a) MNIST and (b) CIFAR-10 datasets. Each subplot shows how Spearman’s ρ , Pearson’s r , and Euclidean distance change as the number of discretization levels increases.

A.3.6 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.3.7 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 the coarse labels. For SVHN, the objective is to predict the cropped digits. For CUB dataset, we use the version available at <https://huggingface.co/datasets/>

Table 7: Aggregated results of MRR CDist Sym.

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(L2)	0.046 ± 0.013	0.105 ± 0.031	0.179 ± 0.173	0.187 ± 0.141	0.04 ± 0.021
RelGeo(Diet)	0.252 ± 0.189	0.278 ± 0.211	0.462 ± 0.148	0.433 ± 0.212	0.306 ± 0.188

Table 8: Aggregated results of MRR 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(L2)	0.019 ± 0.005	0.046 ± 0.016	0.236 ± 0.08	0.156 ± 0.089	0.013 ± 0.005
RelGeo(Diet)	0.189 ± 0.108	0.241 ± 0.117	0.358 ± 0.126	0.327 ± 0.184	0.131 ± 0.107

birders-project/CUB_200_2011-WDS. Given the relatively small training set, we select 2000 points as the validation set. When reporting aggregated MRR metrics in the tables, we always exclude the diagonal entries as these are generally (close to) 1. For ImageNet-1k, we use the validation set and split it into the final train, val and test sets. Further details can be found in the provided code.

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. We always use the Adam optimizer [Kingma and Ba, 2017].

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

A.4 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. We refer to accuracy as *Accuracy*, symmetricized MRR based on cosine as *MRR Cosine Sym*, symmetricized MRR based on cdist as *MRR CDist Sym*, MRR based on cosine as *MRR Cosine* and MRR based on cdist as *MRR CDist*.

A.4.1 Full results

We provide the heatmaps on the different datasets in Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13.

A.4.2 Other evaluation metrics

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

A.4.3 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 11, Table 12, Table 13 and Table 14. In general, the observation remains: RelGeo(L2) yields good accuracies and RelGeo(Diet) yields good MRRs.

Table 9: Aggregated results of MRR 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(L2)	0.019 ± 0.005	0.04 ± 0.015	0.106 ± 0.11	0.108 ± 0.092	0.012 ± 0.005
RelGeo(Diet)	0.127 ± 0.118	0.151 ± 0.138	0.298 ± 0.141	0.269 ± 0.195	0.123 ± 0.103

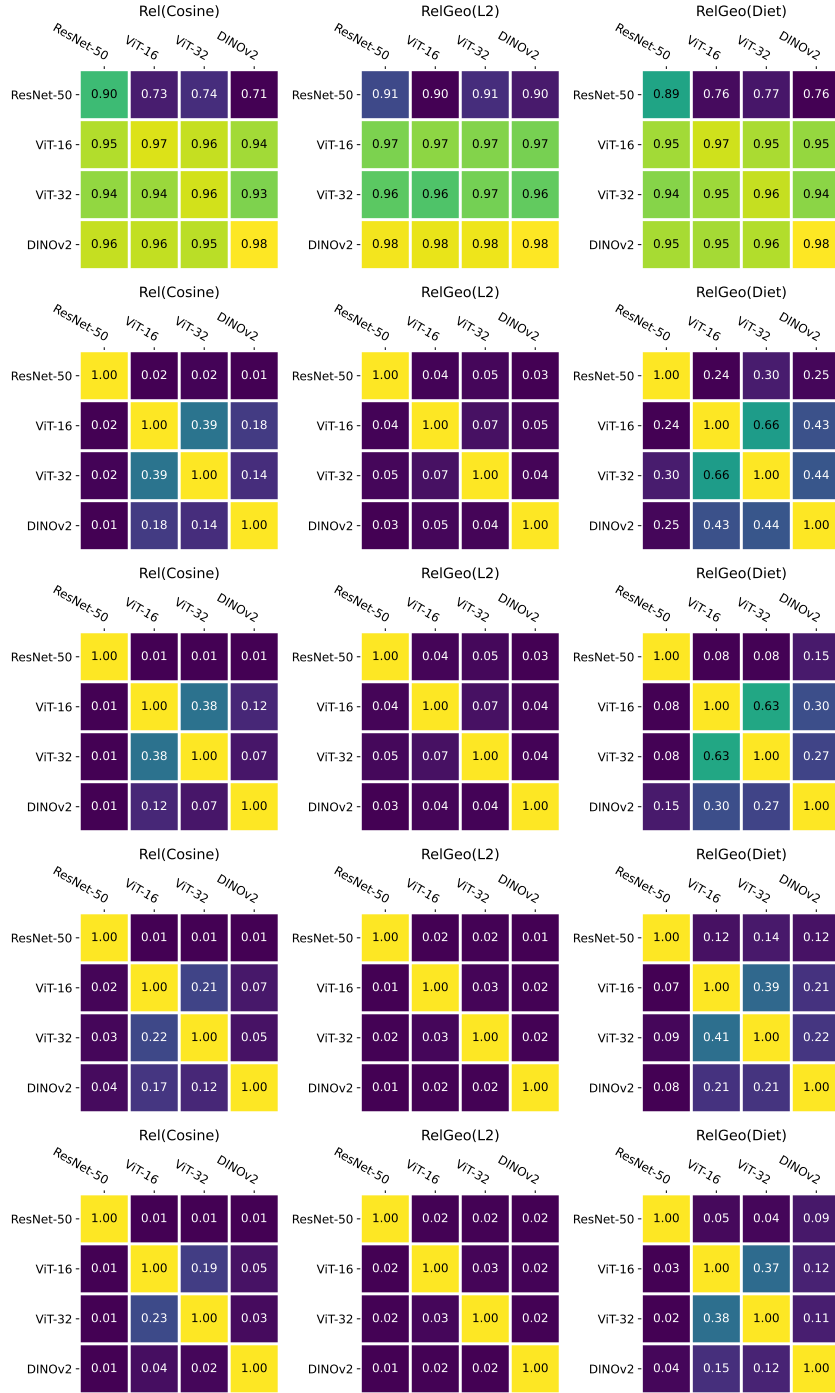


Figure 9: Results on CIFAR-10. From top to bottom: Accuracy, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist

Table 10: Alternatively aggregated results of Accuracy.

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(L2)	0.646 ± 0.209	0.709 ± 0.208	0.72 ± 0.194	0.737 ± 0.209
RelGeo(Diet)	0.529 ± 0.194	0.658 ± 0.229	0.661 ± 0.219	0.668 ± 0.237

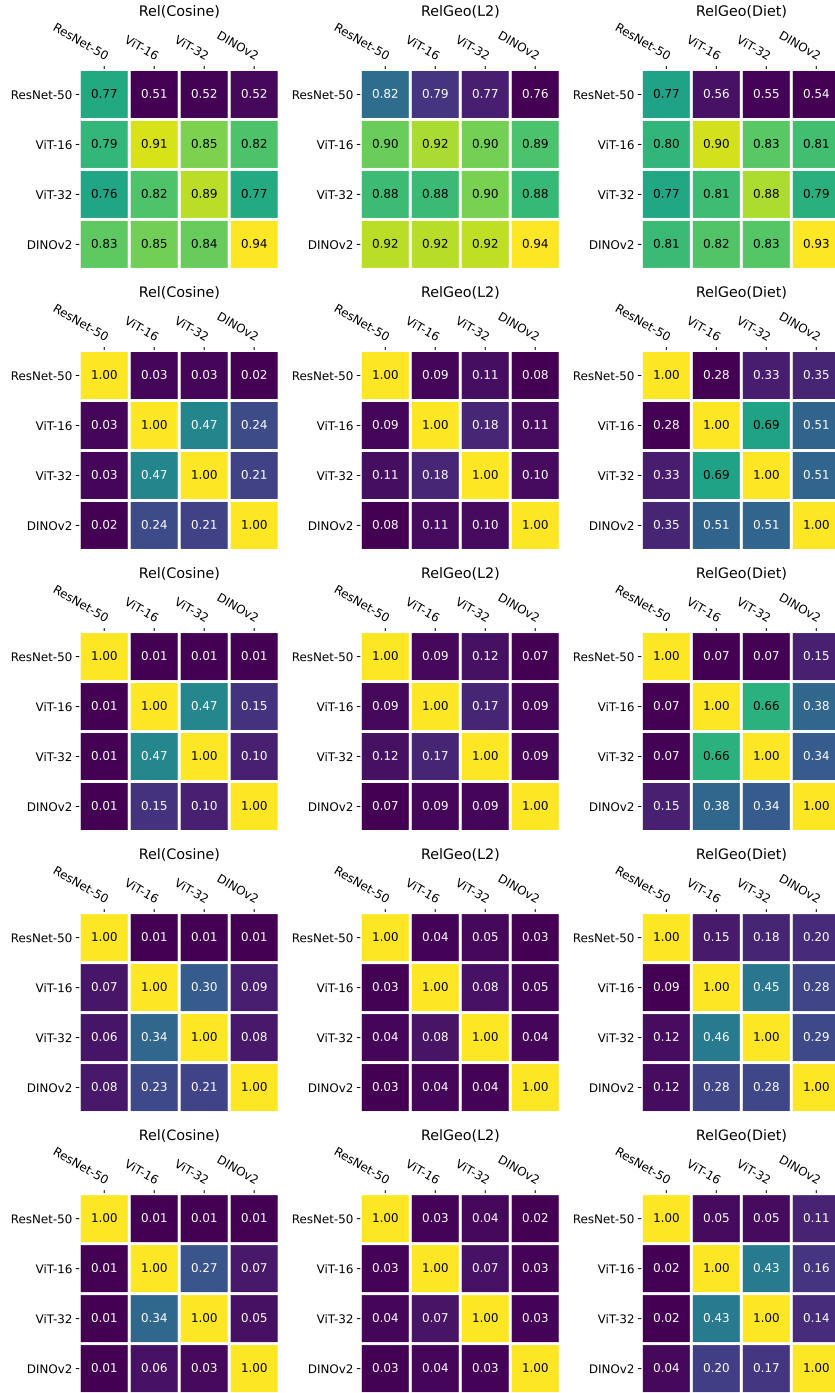


Figure 10: Results on CIFAR-100. From top to bottom: Accuracy, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist

Table 11: Alternatively aggregated results of MRR Cosine Sym.

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(L2)	0.143 ± 0.132	0.197 ± 0.186	0.205 ± 0.189	0.154 ± 0.137
RelGeo(Diet)	0.336 ± 0.143	0.506 ± 0.2	0.526 ± 0.187	0.42 ± 0.106

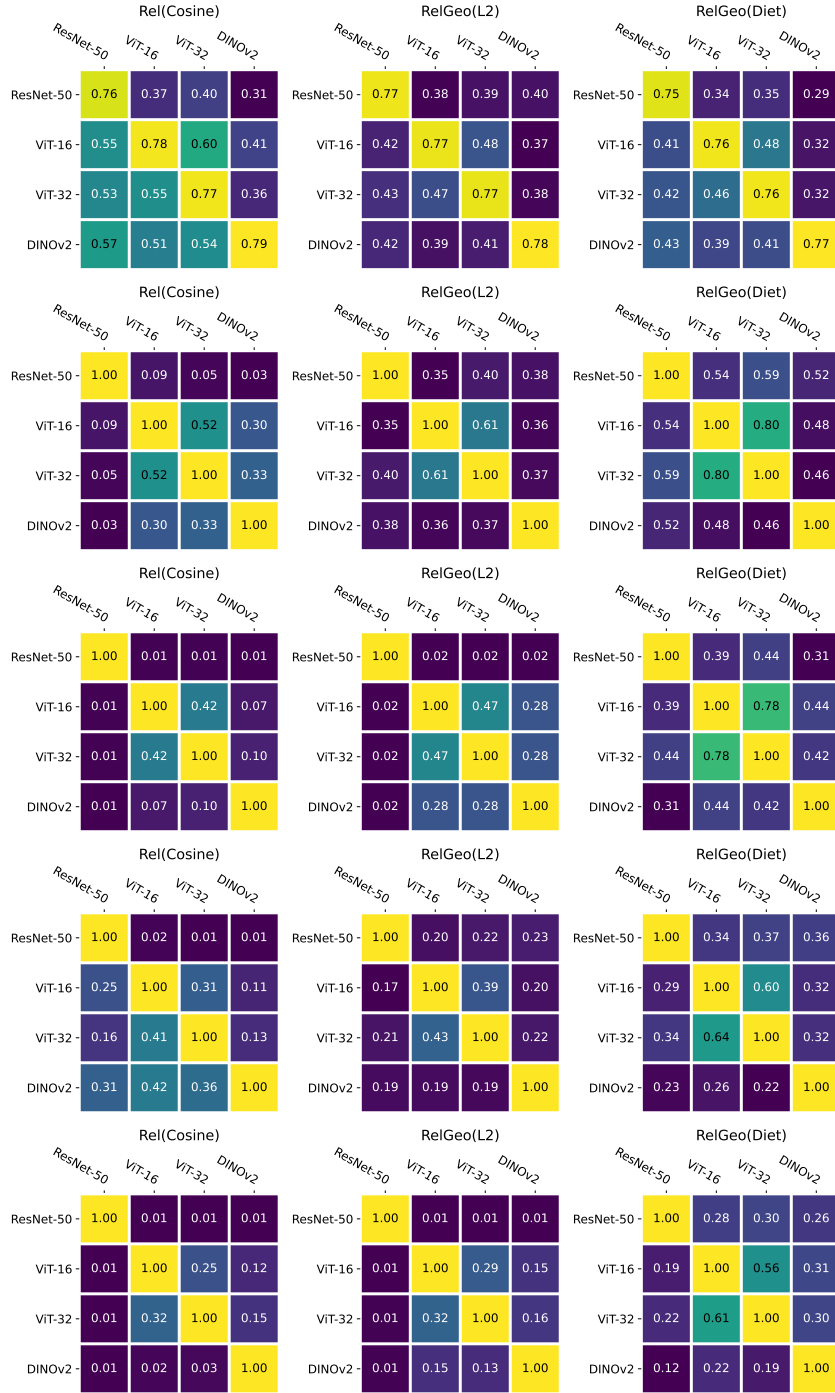


Figure 11: Results on ImageNet-1k. From top to bottom: Accuracy, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist

Table 12: Alternatively aggregated results of MRR CDist Sym.

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(L2)	0.052 ± 0.033	0.144 ± 0.146	0.147 ± 0.146	0.103 ± 0.085
RelGeo(Diet)	0.204 ± 0.13	0.432 ± 0.235	0.437 ± 0.232	0.313 ± 0.108

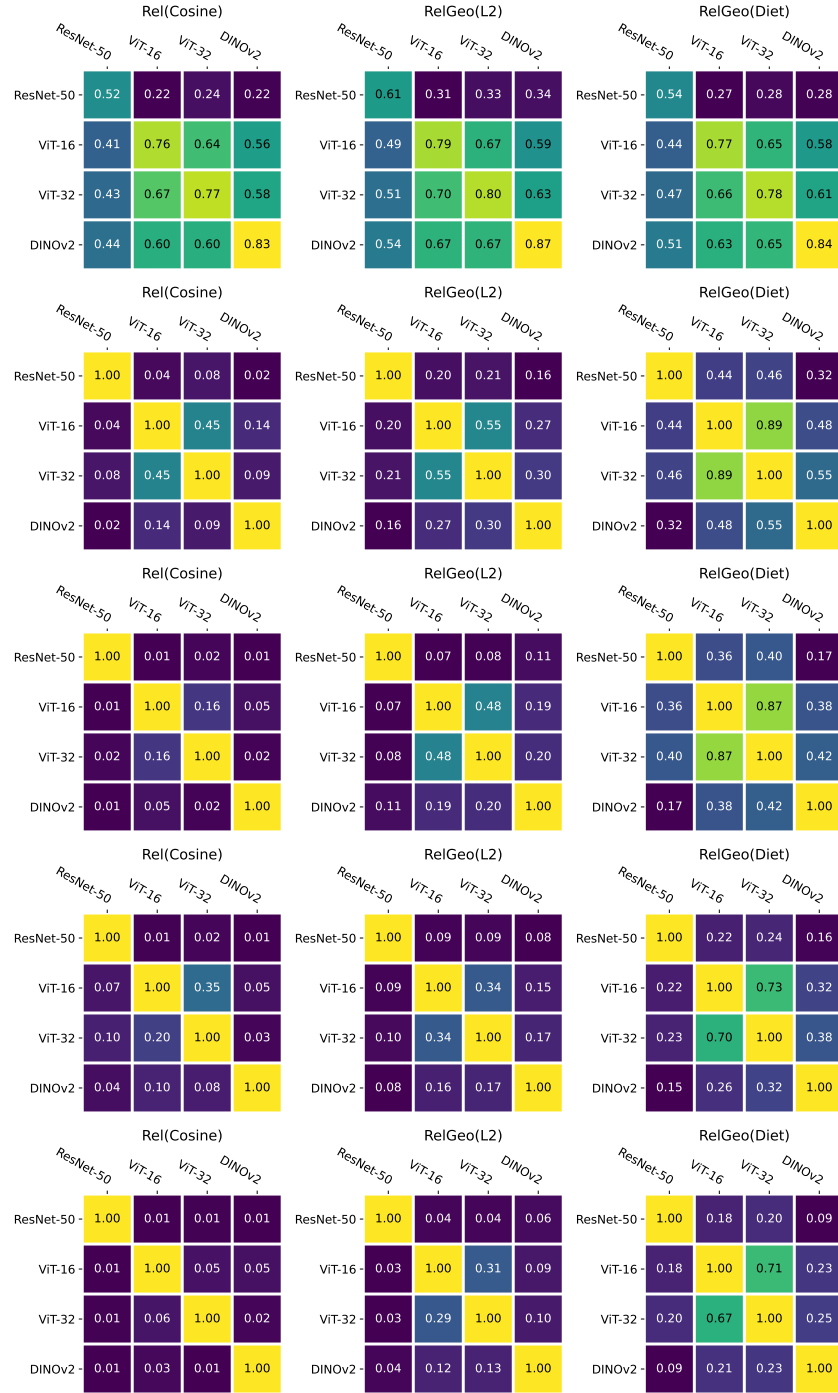


Figure 12: Results on CUB. From top to bottom: Accuracy, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist

Table 13: Alternatively aggregated results of MRR Cosine.

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(L2)	0.074 ± 0.077	0.107 ± 0.118	0.116 ± 0.126	0.079 ± 0.074
RelGeo(Diet)	0.182 ± 0.107	0.299 ± 0.184	0.316 ± 0.182	0.201 ± 0.076

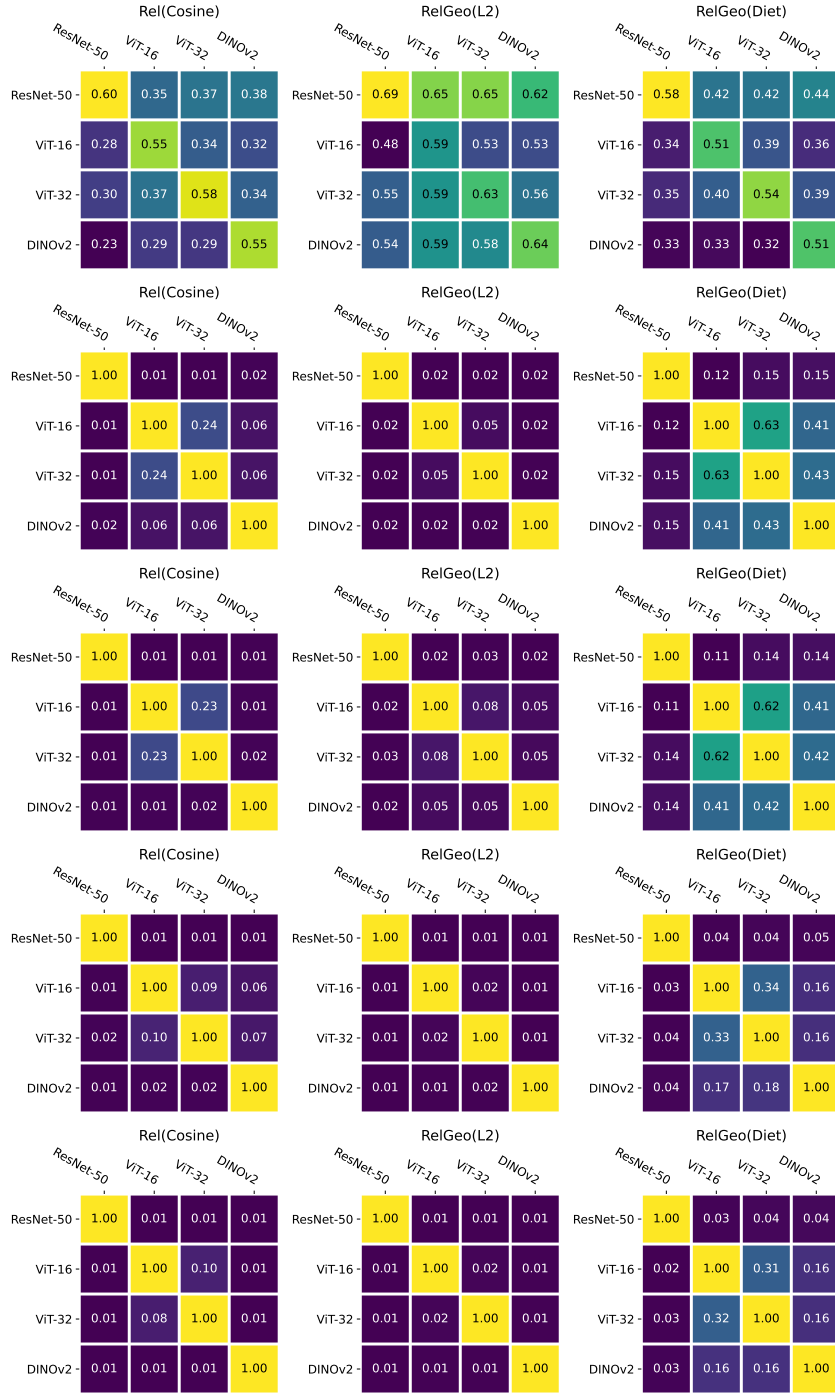


Figure 13: Results on SVHN. From top to bottom: Accuracy, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist

Table 14: Alternatively aggregated results of MRR CDist.

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(L2)	0.023 ± 0.016	0.075 ± 0.096	0.078 ± 0.099	0.051 ± 0.05
RelGeo(Diet)	0.121 ± 0.094	0.253 ± 0.193	0.258 ± 0.194	0.143 ± 0.065

A.4.4 Number of anchors

We investigate the impact of the number of anchors. The results are shown in Figure 14 and Figure 15. The general conclusion that RelGeo(L2) is good in terms of accuracies, RelGeo(Diet) is good in terms of MRRs persist with varying number of anchors.

A.4.5 Number of Diet points

We analyze the impact of the number of Diet points. The results are shown in Figure 16. The performances of RelGeo(Diet) improve as the number of diet points become larger.

A.4.6 Number of discretization steps

We analyze the impact of the number of discretization steps on RelGeo(L2) and RelGeo(Diet) and provide the results in Figure 17 and Figure 18. The performances do not vary much depending on the discretization steps, though using multiple steps seems to help.

A.4.7 Diet augmentation strengths

We analyze the impact of different data augmentation strengths on RelGeo(Diet). The results are shown in Figure 19. Similar to the observations in terms of self-supervised learning [Ibrahim et al., 2024], RelGeo(Diet) benefits from stronger data augmentations.

A.4.8 Anchor selection scheme

As discussed in Moschella et al. [2023], there are different ways of choosing the anchors. In the main paper, we consider the case where the anchors are selected uniformly at random, referred to as *uniform*. There are other choices as well, e.g. using farthest point sampling, referred to as *fps*, and using as anchors the data point close to the centroids of K-means clustering, referred to as *kmeans*.

Here we additionally report the results for *fps* and *kmeans*. Since we need to align multiple models, in practice we use the selection mechanism to select a fixed number of anchors based on the representations of each model, and combine them while employing random subsampling to obtain the final anchors of a given number.

The experimental results for *fps* and *kmeans* are shown in Figure 20 and Figure 21, respectively.

A.4.9 Multimodal

In the main paper we reported results based on MRR Cosine Sym. Below we show the full experimental results in Figure 22, and report aggregated results in Table 15.

Table 15: Average MRR results for different methods across different datasets. Relative representations pulling back from diet decoder (RelGeo(Diet)) consistently provides better retrievals.

Method	MRR Cosine Sym	MRR CDist Sym	MRR Cosine	MRR CDist
Rel(Cosine) [Moschella et al., 2023]	0.298 $\pm$ 0.395	0.293 $\pm$ 0.389	0.283 $\pm$ 0.382	0.252 $\pm$ 0.373
RelGeo(Diet)	0.413 $\pm$ 0.353	0.384 $\pm$ 0.359	0.317 $\pm$ 0.372	0.302 $\pm$ 0.378

A.4.10 Running times

We report running times for autoencoder experiments with an NVIDIA 3080 Ti GPU and vision foundational model experiments with an NVIDIA A100 GPU.

Times to train the models For autoencoder experiments, training the autoencoders is not expensive, as it merely takes around 20 minutes on an RTX3080Ti for 50 epochs.

For vision foundation models, the employed pretrained backbones are typically computationally costly to train. For instance, Oquab et al. [2024] reported that training DINOv2 ViT-L/14 on ImageNet-22k using 96 A100-80GB GPUs takes approximately 3.3 days. In contrast, training the decoders is much faster. We report the running times to train the decoders in Table 16, where for each setting we

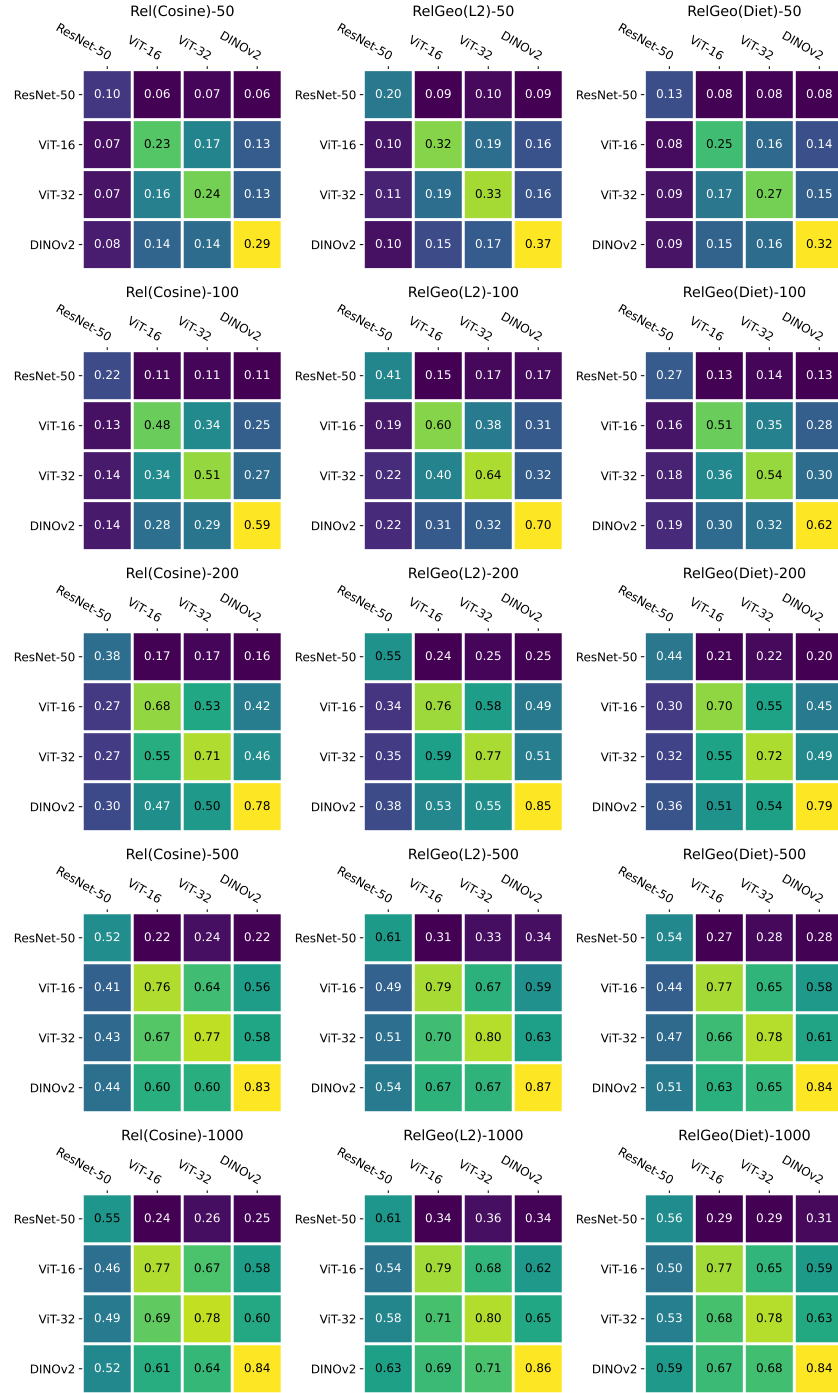


Figure 14: Accuracies on CUB with varying number of anchors. From top to bottom: 50, 100, 200, 500, 1000 anchors.

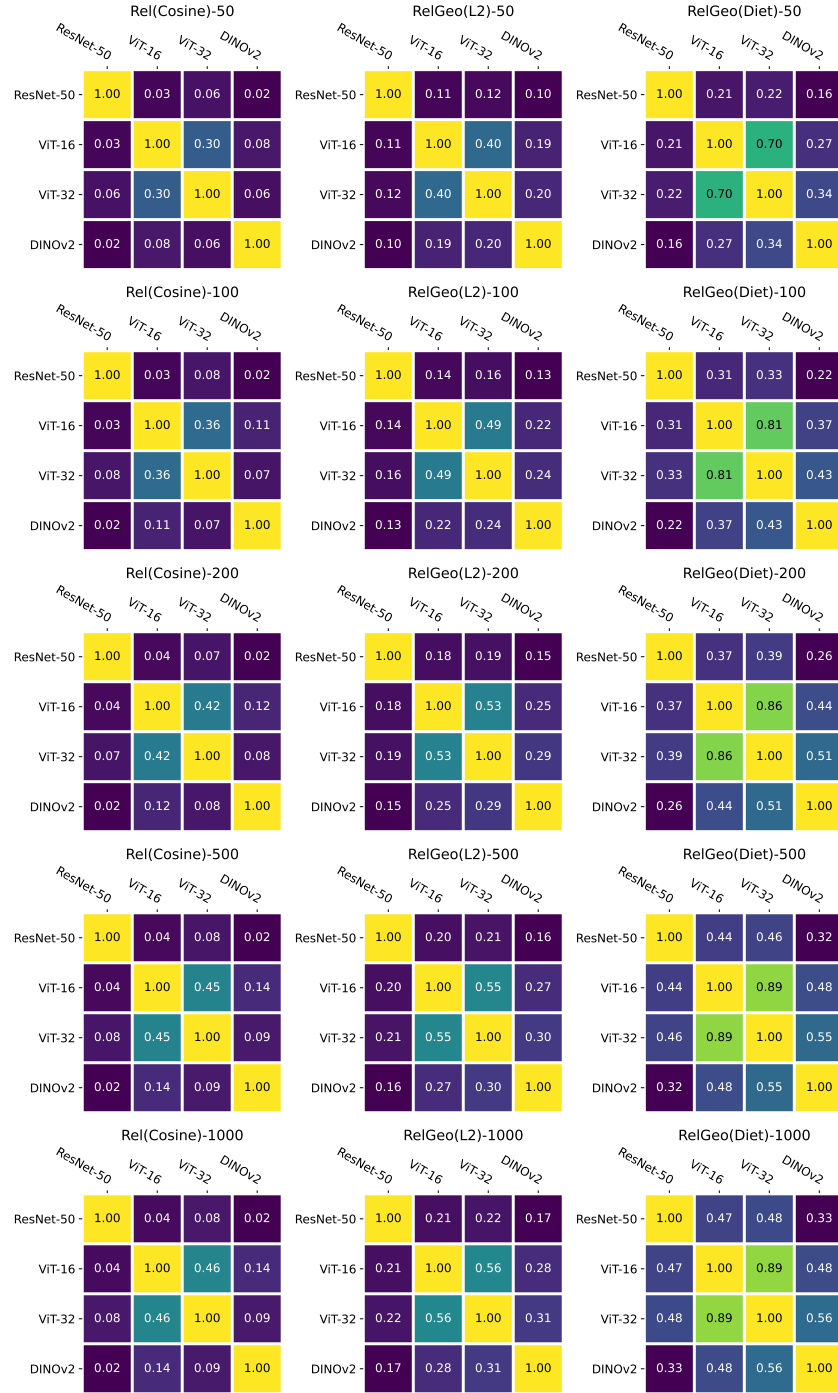


Figure 15: MRR Cosine Sym on CUB with varying number of anchors. From top to bottom: 50, 100, 200, 500, 1000 anchors.

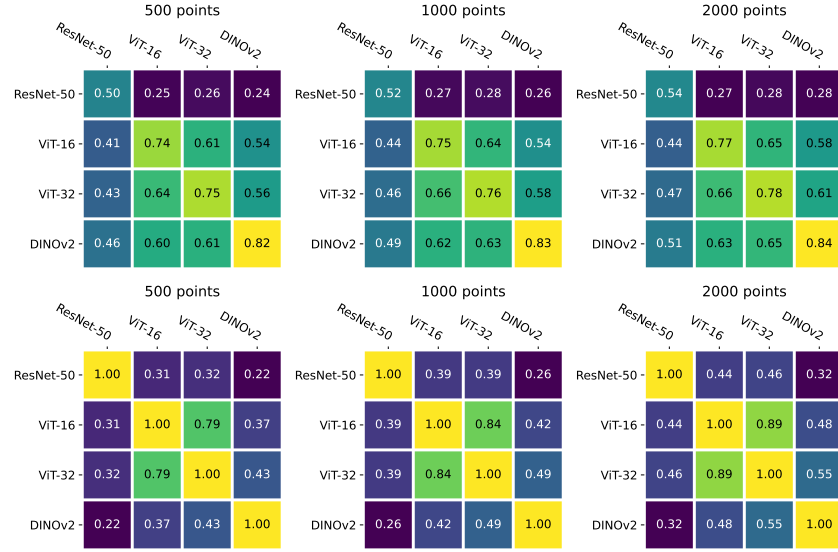


Figure 16: Results of RelGeo(Diet) on CUB with varying number of diet points. Top: accuracies; bottom: MRR Cosine Sym.

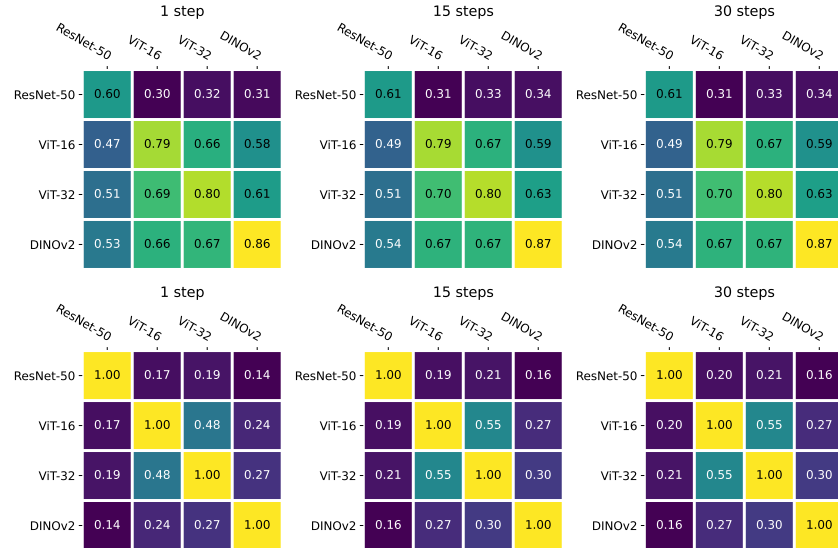


Figure 17: Results of RelGeo(L2) on CUB with varying number of discretization steps. Top: accuracies; bottom: MRR Cosine Sym.

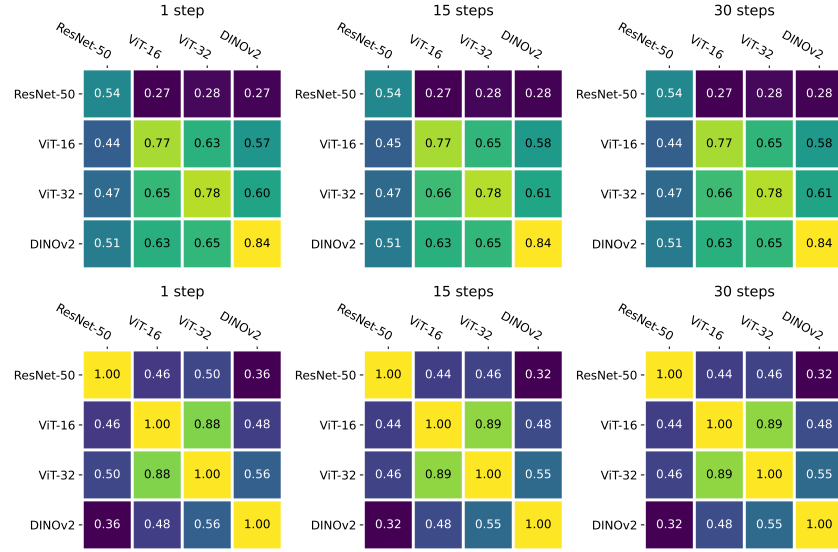


Figure 18: Results of RelGeo(Diet) on CUB with varying number of discretization steps. Top: accuracies; bottom: MRR Cosine Sym.

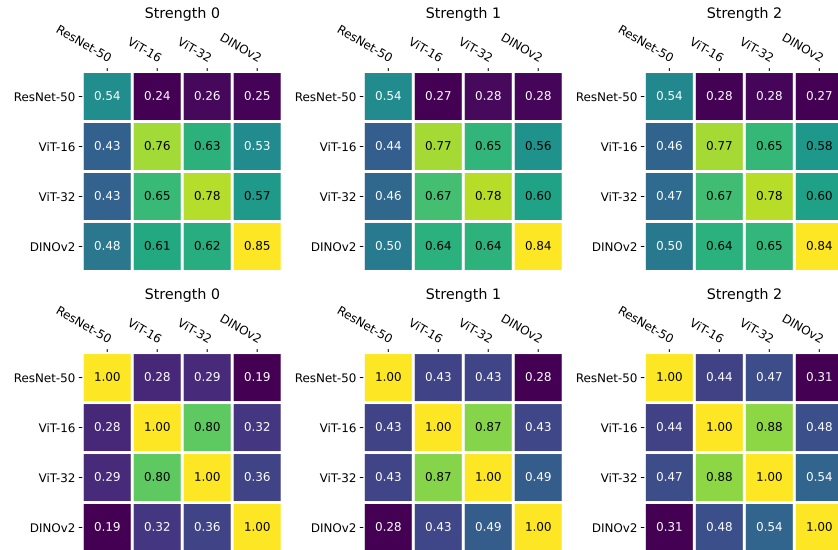


Figure 19: Results of RelGeo(Diet) on CUB with varying diet augmentation strengths. Results with strength 3 can be seen above. Top: accuracies; bottom: MRR Cosine Sym.

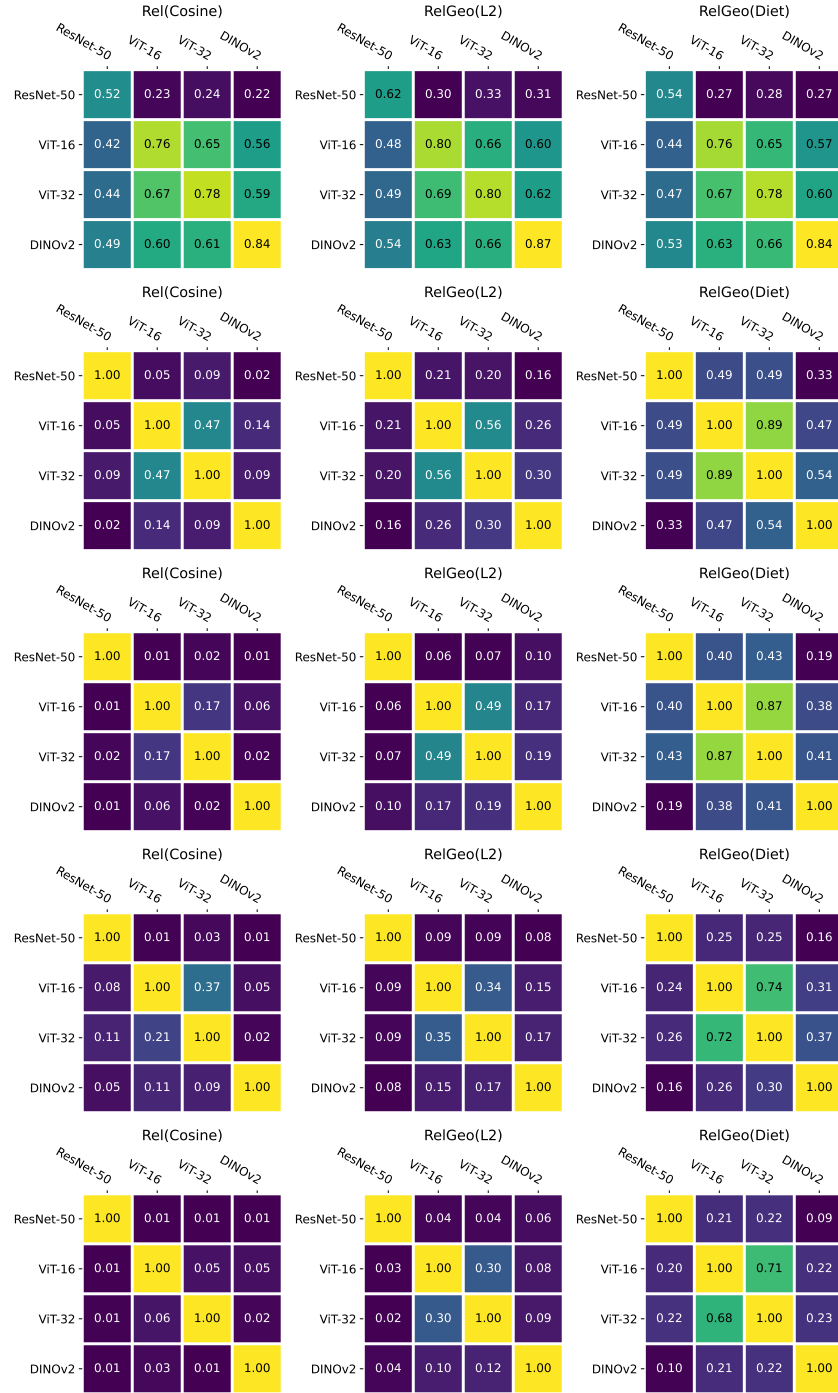


Figure 20: Results using the *fps* scheme to select the anchors. From top to bottom: accuracies, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist.

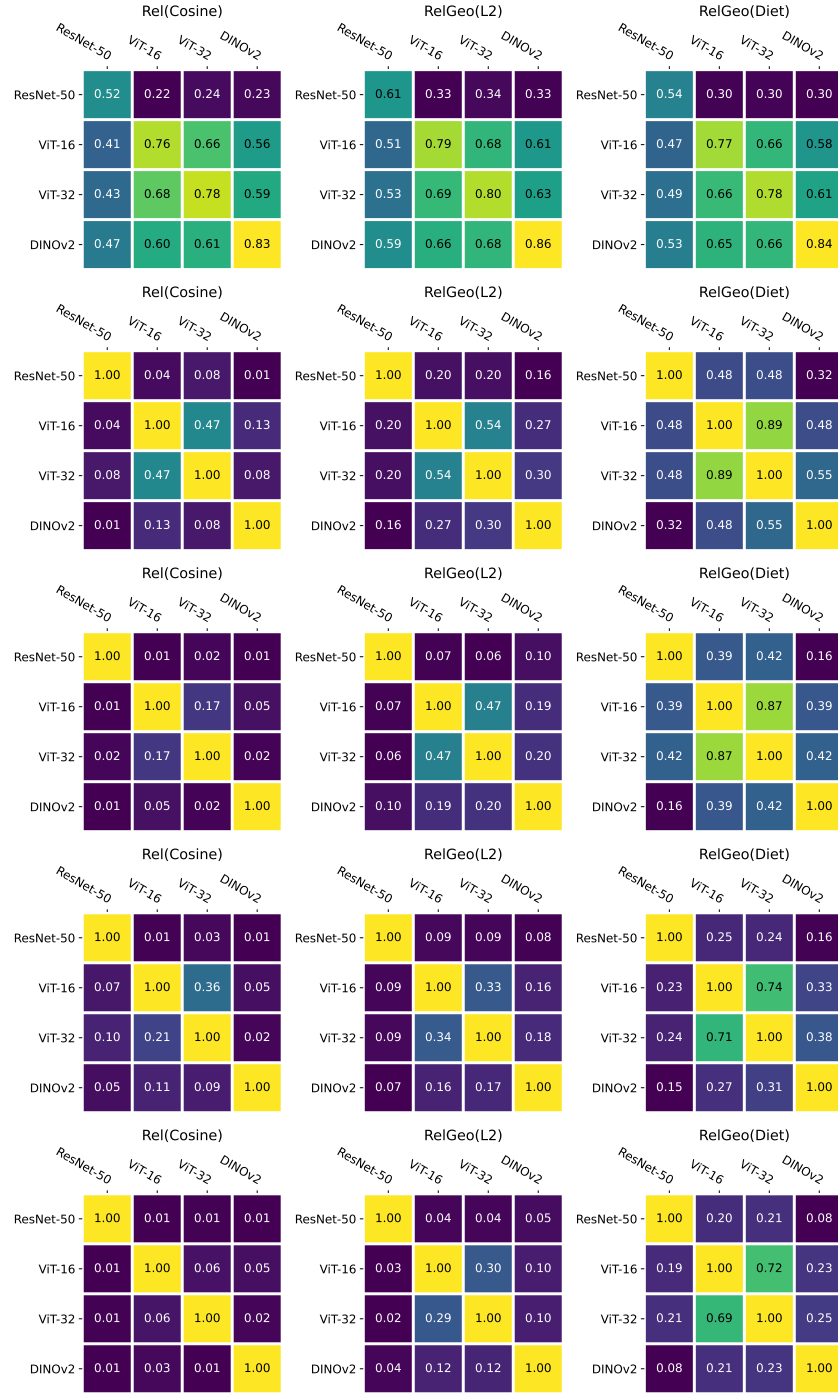


Figure 21: Results using the *kmeans* scheme to select the anchors. From top to bottom: accuracies, MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist.

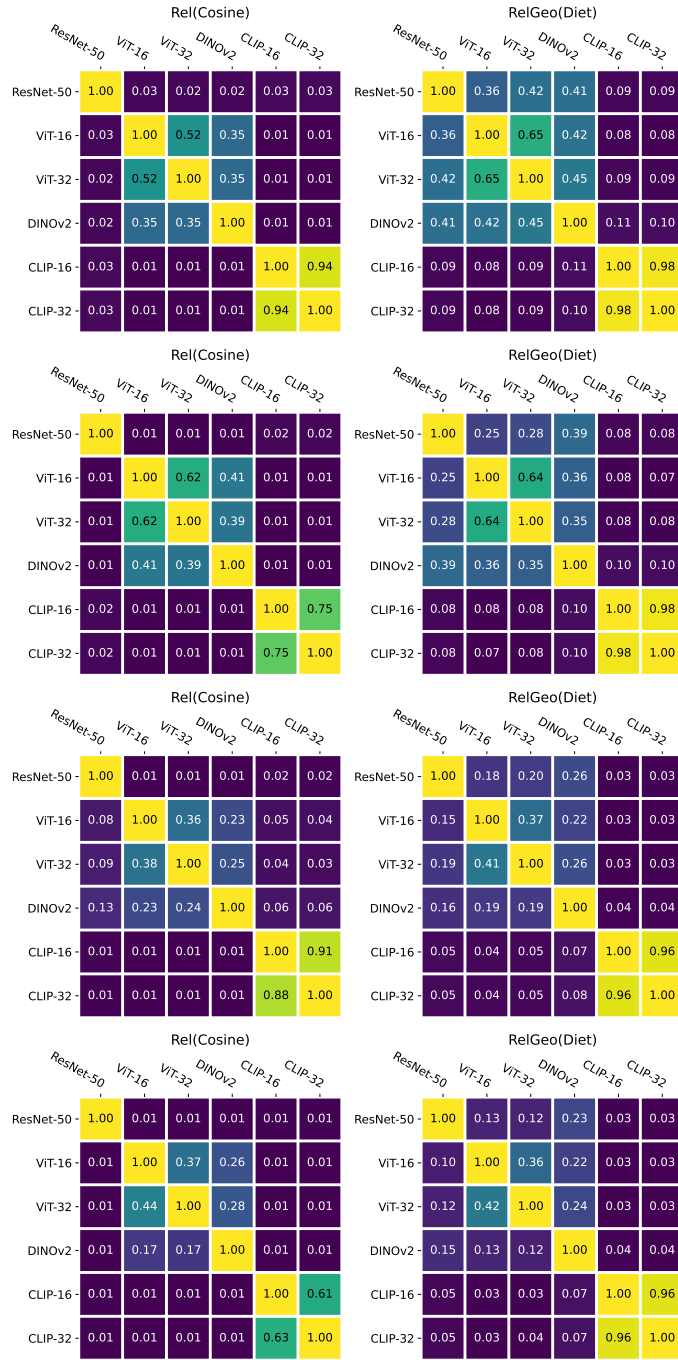


Figure 22: Stitching results for multimodal scenario. From top to bottom: MRR Cosine Sym, MRR CDist Sym, MRR Cosine, MRR CDist.

average over the different models to obtain uncertainty estimates. We remark that the exact running times depend heavily on implementation details, while currently we focus on correctness instead of the speed, and the running times could possibly be improved with better implementations.

Table 16: Times in seconds for training the decoders.

Decoder	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
Abs	10.467 ± 0.765	10.71 ± 0.655	8.804 ± 0.581	1.593 ± 0.615	15.57 ± 0.843
Diet(0)	3.462 ± 0.03	3.501 ± 0.045	3.556 ± 0.018	3.509 ± 0.028	3.506 ± 0.044
Diet(1)	718.818 ± 216.179	730.446 ± 226.91	1651.49 ± 211.037	1937.656 ± 192.894	705.929 ± 202.844
Diet(2)	735.144 ± 219.211	739.671 ± 222.36	2034.677 ± 276.309	2428.942 ± 196.388	727.477 ± 206.929
Diet(3)	771.299 ± 213.022	767.453 ± 222.62	2645.914 ± 260.069	3201.588 ± 242.339	754.965 ± 202.021

Times to obtain the representations We first investigate the running times - accuracy tradeoff of RelGeo representations on CUB dataset, where we vary the number of anchors and monitor the times to obtain the representations and the qualities of the resulting representations. We report the results on autoencoders in Table 17 and the results on vision foundation models in Table 18, Table 19 and Table 20, respectively.

We then investigate the times to evaluate the representations across different datasets and report the results in Table 21, under the same experimental settings as reported in the main paper.

A.4.11 RelGeo(Fisher)

We additionally report the results of relative geodesic representations based on another choice of Rimennian metric, RelGeo(Fisher), which pulls back the Fisher-Rao metric from the classification heads’ output probabilities on the different datasets. We report the aggregated results in Table 22, and the results on the individual datasets in Figure 23, Figure 24, Figure 25, Figure 26 and Figure 27. RelGeo(Fisher) often results in higher accuracies and lower MRRs; we hypothesize that this is due to the Neural Collapse phenomenon [Kothapalli, 2023] observed in well-trained neural networks.

Table 17: Running time / accuracy tradeoff of RelGeo(L2).

Num Anchors	Time (s) $\pm$ Std	MRR
2	0.8480 ± 0.0407	0.0168
3	0.8348 ± 0.0367	0.0807
5	0.8377 ± 0.0435	0.3503
8	0.9450 ± 0.0285	0.7004
10	1.0969 ± 0.0297	0.8384
15	1.2853 ± 0.0264	0.9296
20	1.6160 ± 0.0188	0.9616
25	1.8548 ± 0.0218	0.9868
50	3.3311 ± 0.0286	0.9981
100	6.2122 ± 0.0318	0.9986
300	17.6494 ± 0.0653	0.9982
500	29.1925 ± 0.0806	0.9986

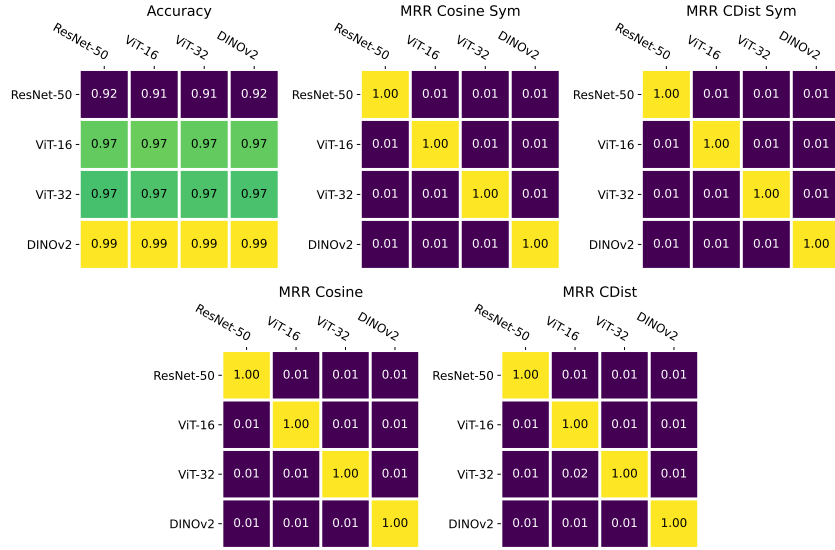


Figure 23: Results of RelGeo(Fisher) on CIFAR-10.

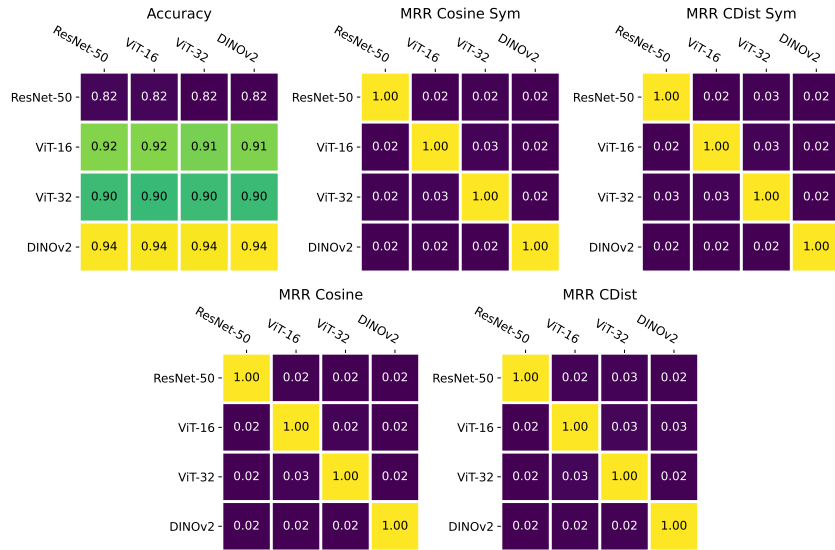


Figure 24: Results of RelGeo(Fisher) on CIFAR-100.

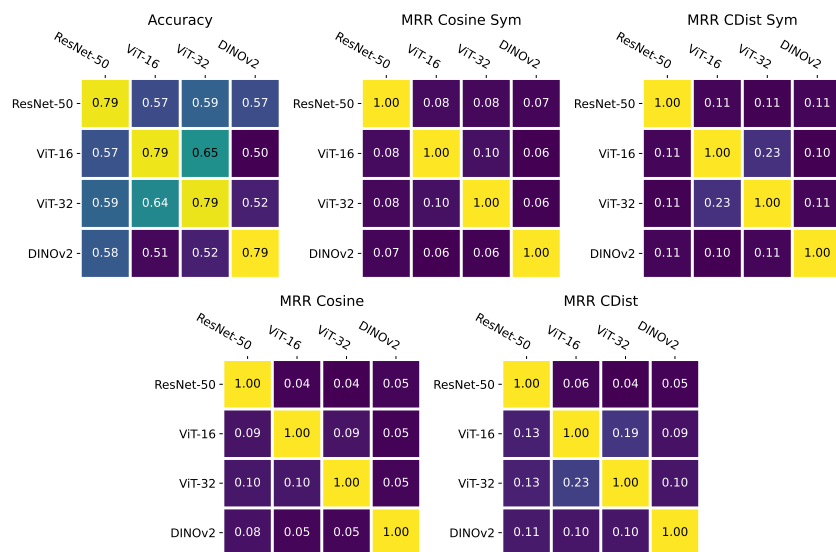


Figure 25: Results of RelGeo(Fisher) on ImageNet-1k.

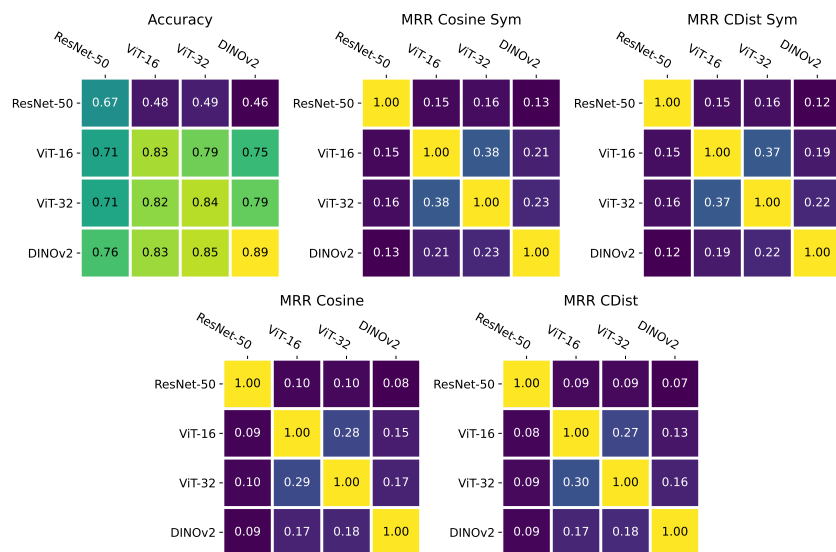


Figure 26: Results of RelGeo(Fisher) on CUB.

Table 18: Running time / accuracy tradeoff of Rel(Cosine).

Num Anchors	Time (s)	Accuracy
200	0.088 ± 0.14	0.425 ± 0.189
500	0.175 ± 0.053	0.531 ± 0.188
1000	0.062 ± 0.091	0.559 ± 0.179

Table 19: Running time / accuracy tradeoff of RelGeo(L2).

Num Anchors	Time (s)	Accuracy
200	8.004 ± 3.304	0.5 ± 0.184
500	21.297 ± 8.774	0.595 ± 0.163
1000	47.218 ± 19.486	0.619 ± 0.154

Table 20: Running time / accuracy tradeoff of RelGeo(Diet).

Num Anchors	Time (s)	Accuracy
200	7.274 ± 3.28	0.459 ± 0.177
500	19.427 ± 8.771	0.559 ± 0.171
1000	43.108 ± 19.502	0.585 ± 0.163

Table 21: Times in seconds of Rel(Cosine), RelGeo(L2) and RelGeo(Diet) for generating the representations.

Method	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
Rel(Cosine) [Moschella et al., 2023]	0.071 ± 0.113	0.184 ± 0.065	0.085 ± 0.124	0.05 ± 0.071	
RelGeo(L2)	85.414 ± 40.61	86.079 ± 40.652	258.788 ± 70.064	139.998 ± 66.509	
RelGeo(Diet)	89.899 ± 40.599	89.902 ± 40.588	155.991 ± 70.136	147.336 ± 66.505	

Table 22: Results of RelGeo(Fisher).

Metric	CIFAR-10	CIFAR-100	ImageNet-1k	CUB	SVHN
Accuracy	0.959 ± 0.026	0.894 ± 0.043	0.623 ± 0.103	0.729 ± 0.133	0.625 ± 0.033
MRR Cosine Sym	0.012 ± 0.0	0.021 ± 0.003	0.075 ± 0.014	0.211 ± 0.082	0.034 ± 0.01
MRR CDist Sym	0.012 ± 0.001	0.025 ± 0.004	0.13 ± 0.046	0.201 ± 0.08	0.028 ± 0.008
MRR Cosine	0.012 ± 0.001	0.019 ± 0.003	0.066 ± 0.022	0.152 ± 0.069	0.011 ± 0.002
MRR CDist	0.012 ± 0.001	0.023 ± 0.005	0.111 ± 0.052	0.144 ± 0.073	0.011 ± 0.003



Figure 27: Results of RelGeo(Fisher) on SVHN.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Main claims have been listed in the introduction and they have been explained in theoretical aspects (see Sec. 3) and have been supported by experimental results (Sec. 4).

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations in our introduced approach has been discussed in Sec. 5, including the possible limitations with scalability and computational efficiency.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Theoretical background and notation are clearly explained in detail in Sec. 3.1.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide the pseudocode for our approach in Sec. 3, as well as additional implementation details (such as architectural details and hyperparameters) in Appendix A.3. All the models and the datasets used in the experiments have been cited. Experimental settings have also been explained in Sec. 4, 4.2, including details such as number of anchors used in each experiment.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the datasets used in the experiments have open access and they been cited accordingly in the paper. Codebase is not included in the main paper (it is available in the supplementary material), and the architectural details of the implemented models, such as classification head and decoders, are explained in Appendix A.3.2. Other pretrained models are explained clearly (including the hyperparameter choices) and cited, partly in the main paper, and mostly in Appendix A.3.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Some experimental results are provided in the main paper, while more are provided in Appendix A.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our result tables report both the means and the standard deviations. Further details on the statistical properties of the performances of the methods are provided in Appendix A.4, where we provide the full results and results under alternative aggregation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Details on the compute resources are provided in Appendix A.3.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We closely follow the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Our work studies geometric concepts in distinct neural network models, thus has the potential to advance our understandings of deep learning in general. We remark that our work thus also suffers from the potential negative societal impacts of deep learning, as the trained models might be used for dubious purposes.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release such data or models. We largely focus on using open source data and open source models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We provide some information in the main paper, and more in the Appendix A.3.2.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Concerning code, our code is provided in the Supplement and will be openly available upon acceptance. Concerning data, we focus on open source datasets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: The paper does not involve research with human subjects, therefore it is not relevant.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve research with human subjects, therefore the question is not relevant.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method of the paper does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.