Latent Functional Maps

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2024

Abstract

Neural models learn data representations which lie on low dimensional manifolds, yet modelling the relation between these representational spaces is an ongoing challenge. By integrating spectral geometry principles into neural modeling, we show that this problem can be better addressed in the functional domain, mitigating complexity, while enhancing interpretability and performances on downstream tasks. To this end, we introduce a multi-purpose framework to the representation learning community which allows to: (i) compare different spaces in an interpretable way and measure their intrinsic similarity; (ii) find correspondences between them, both in unsupervised and weakly supervised settings, and (iii) to effectively transfer representations between distinct spaces. We validate our framework on various applications, ranging from stitching to retrieval tasks, demonstrating that latent functional maps can serve as a swiss-army knife for representation alignment.

1. Introduction

Recent studies have shown that neural models often develop similar representations when exposed to similar stimuli, both in biological [10, 17] and artificial settings [15, 28, 29]. Notably, internal representations of distinct models can often be aligned through a linear transformation [38, 47] (e.g. when subject to different initializations). This indicates a level of consistency in how NNs process information, showing the importance of characterizing these internal representations and their geometric relation. In this paper, we shift our focus from characterizing relationships between samples in distinct latent spaces to modelling a map between function spaces defined on these latent manifolds. We leverage the framework of *functional maps* [33], applying it for the first time to the field of representation learning. Functional maps represent correspondences between function spaces on different manifolds: in this setting, many difficult constraints can be easily manipulated and expressed compactly [35]. For instance, as shown in Figure 1, the mapping in the functional space (C) becomes a linear map with a sparse structure. Our contributions can be listed as follows: (i) We introduce the framework of Latent Functional Maps as a way to model the relation between distinct representational spaces of neural models. (ii) We show that LFM allows us to find correspondences between representational spaces, both in weakly supervised and unsupervised settings, and to transfer representations across distinct models. (iii) We showcase LFM capabilities as a meaningful and interpretable similarity measure between representational spaces. (iv) We validate our findings in retrieval and stitching tasks across different models, modalities and datasets, demonstrating that LFMs can lead to better performance and sample efficiency than other methods.

© .

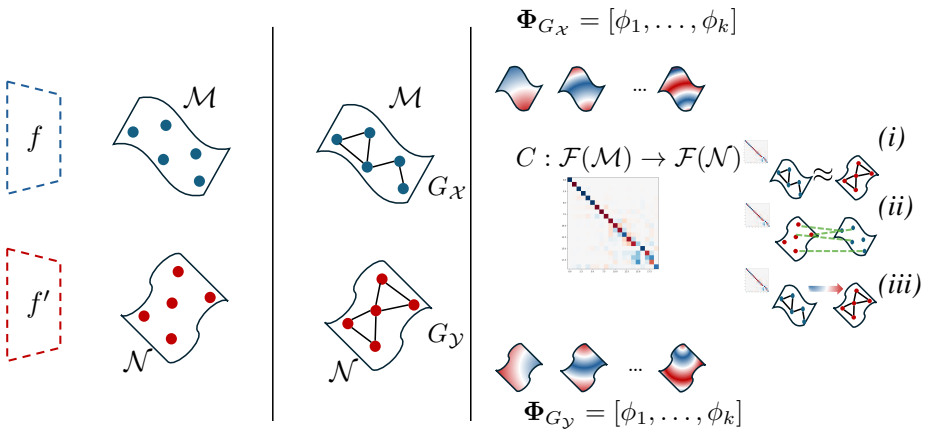


Figure 1: **Framework overview:** given two representational spaces $\mathcal{X}$, $\mathcal{Y}$ their samples lie on two manifold $\mathcal{X}$, $\mathcal{Y}$, which can be approximated with the KNN graphs $G_\mathcal{X}$,$G_\mathcal{Y}$; We can optimize for a *latent functional map* C between the eigenbases of the graphs. This map serves as a map between functions defined on the two manifolds and can be leveraged for comparing representational spaces, solving correspondence problems, and transferring information between the spaces.

2. Method

We provide the basic notions to understand the framework of functional maps applied to manifolds in the Appendix A.

Setting We consider deep neural networks $f := f_1 \circ f_2 \circ ...f_n$ where each layer f_i is associated to a representational space $\mathcal{X}$ corresponding to the image of f_i. We assume that elements $x \in \mathcal{X}$ are sampled from a latent manifold $\mathcal{M}$. Considering pairs of spaces $(\mathcal{X}, \mathcal{Y})$, and corresponding manifolds $\mathcal{M}, \mathcal{N}$ our objective is to characterize the relation between them by mapping the space of functions $\mathcal{F}(\mathcal{M})$ to $\mathcal{F}(\mathcal{N})$. Our framework is depicted in Figure 1. In the following, we will start by approximating $\mathcal{X}$ from a sample estimate, building a graph in the latent space.

Latent Functional Maps We model each space using a subset of training samples $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ and build a k-NN graphs G_X and G_Y from these samples, respectively with a given distance metric (for deatils about the graph construction see Appendix C.1.1). For each graph, we compute the graph Laplacian $\mathcal{L}_G$ and derive the first k eigenvalues Λ_G and eigenvectors $\mathbf{\Phi}_G = [\phi_1, \ldots, \phi_k]$, which serve as the basis for the function space defined on the latent spaces.

Given the set of corresponding functions $\mathbf{F}_{G_X} = [f_1^{G_X}, \ldots, f_{n_f}^{G_X}]$ and $\mathbf{F}_{G_Y} = [f_1^{G_Y}, \ldots, f_{n_f}^{G_Y}]$, we consider the optimization problem defined in Equation 2 and incorporate regularizers for Laplacian and descriptor operator commutativity, as defined in [31]:

$$\arg\min_{\mathbf{C}} ||\mathbf{C}\hat{\mathbf{F}}_{G_X} - \hat{\mathbf{F}}_{G_Y}||^2 + \alpha\rho_\mathcal{L}(\mathbf{C}) + \beta\rho_f(\mathbf{C}) \tag{1}$$

where $\hat{\mathbf{F}}_G = \mathbf{\Phi}_G^T \mathbf{F}_G$ are the spectral coefficients of the functions $\mathbf{F}_G$, $\rho_\mathcal{L}$ and ρ_f are the Laplacian and descriptor operator commutativity regularizers respectively. We specify how we compute the regularizers in Appendix C. As a set of corresponding functions, we use the geodesic distance functions computed from a point $x \in X$ to all other points in X, where x is a point for which we

know the corresponding point $y \in Y$ in the other latent space. Once we have solved the optimization problem defined in Equation 1, we refine the resulting functional map $\mathbf{C}$ using the algorithm proposed by [24].

LFMs as a similarity measure Once computed, the functional map $\mathbf{C}$ can serve as a measure of similarity between spaces. The reason is that for isometric transformations between manifolds, the functional map is volume preserving (see Thm 5.1 in [33]), and this is manifested in orthogonal $\mathbf{C}$. By defining the inner product between functions $h_1, h_2 \in \mathcal{F}(M)$ as $\langle h_1, h_2 \rangle = \int_{\mathcal{M}} h_1(x) h_2(x) \mu(x)$, it holds that $\langle h_1, h_2 \rangle = \langle \hat{h}_1, \hat{h}_2 \rangle$ when the map preserves the local area, where $\hat{h}$ denotes the functional representation of h. In other words, when the transformation between the two manifolds is an isometry, the matrix $\mathbf{C}^T\mathbf{C}$ will be diagonal. By measuring the ratio between the norm of the off-diagonal elements of $\mathbf{C}^T\mathbf{C}$ and the norm of its diagonal elements, we can define a measure of similarity $sim(X, Y) = 1 - \frac{||\text{off}((\mathbf{C}^T\mathbf{C})||_F}{||\text{diag}(\mathbf{C}^T\mathbf{C})||_F}$. Furthermore, this quantity is interpretable; the first eigenvector of $\mathbf{C}^T\mathbf{C}$ can act as a signal to localize the area of the target manifold where the map has higher distortion [34].

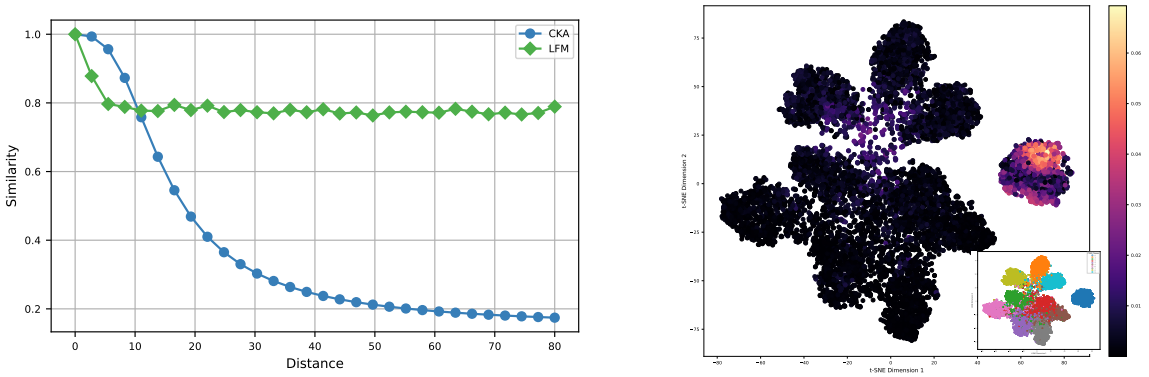
Transfering information with LFM The functional map computed between two latent spaces can be utilized in various ways to transfer information from one space to the other. In this paper, we focus on two methods: (i) Expressing arbitrary points in the latent space as distance function on the graph and transferring them through the functional domain (see C.1.2 for details); (ii) Obtaining a point-to-point correspondence between the representational spaces from the LFM, starting from none to few known pairs, and leverage off-the-shelf methods to learn a transformation between the spaces (see C.1.3 for details). Additional strategies could be explored in future work.

3. Experiments

Analysis We demonstrate the benefits of using latent functional maps for comparing distinct representational spaces, using the similarity metric defined in Section 2. In Appendix E.1, we compare the LFM similarity with CKA and show that our LFM-based similarity measure behaves correctly as CKA does. While CKA (Centered Kernel Alignment) is a widely used similarity metric in deep learning, recent research by [8] has shown that it can produce unexpected or counter-intuitive results in certain situations. Specifically, CKA is sensitive to transformations that preserve the linear separability of two spaces, such as local translations. Our proposed similarity measure is robust to these changes and demonstrates greater stability compared to CKA.

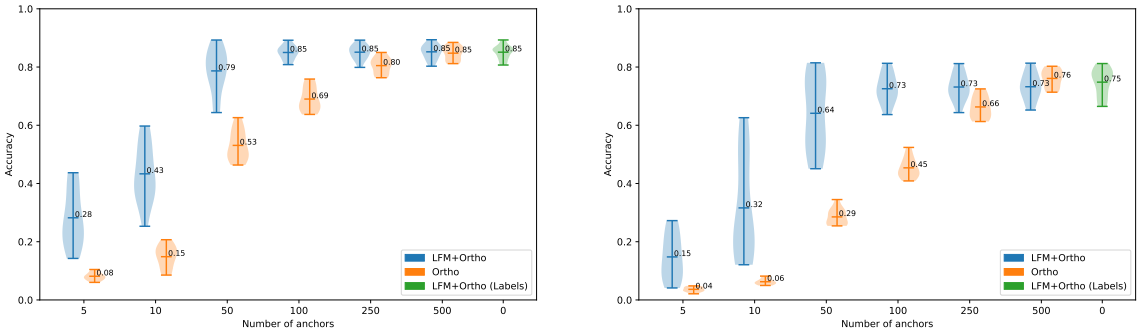
Experimental setting. We compute the latent representations from the pooling layer just before the classification head for the `CIFAR10` train and test sets. Following the setup in [8], we train a Support Vector Machine (SVM) classifier on the latent representations of the training samples to find the optimal separating hyperplane between samples of one class and others. We then perturb the samples by translating them in a direction orthogonal to the hyperplane, ensuring the space remains linearly separable. We measure the CKA and LFM similarities as functions of the perturbation vector norm, as shown in Figure 2(*b*)subfigure. In the accompanying plot on the right, we visualize the area distortion of the map by projecting the first singular component of the LFM $\mathbf{C}$ into the perturbed space and plotting it on a 2d TSNE [43] projection of the space.

Result Analysis. We start by observing that, when the latent space is perturbed in a way that still preserves its linear separability, it should be considered identical from a classification perspective, as this does not semantically affect the classification task. Figure 2(*b*)subfigure shows that while CKA



(*a*) Localized area distortion (*b*) CKA vs LFM similarity

Figure 2: **Robustness of LFM similarity** *Left:* Similarity scores as a function of perturbation strength: while the CKA baseline degrades, our LFM similarity scores are robust to perturbations that preserve linear separability of the space. *Right:* Visualization of area distortion of the map by projecting the first singular component of the LFM in the perturbed space: the distortion localizes on the samples of the perturbed class, making LFM similarity interpretable.



(*a*) Coarse labelling (*b*) Fine labelling

Figure 3: **Stitching on CIFAR100.**Accuracy performance in stitching between image encoders trained on CIFAR100, comparing orthogonal transformation (Ortho) and LFM+Ortho at varying anchor counts. Also shown is LFM+Ortho (Labels), which uses the dataset labels instead of anchors. Results are presented for (a) coarse and (b) fine-grained labeling, with mean accuracy values reported on each box.

degrades as a function of perturbation intensity, the LFM similarity remains stable to high scores. To understand this difference, we can visualize the area distortion as a function of the samples by projecting the first singular component of $\mathbf{C}$ onto the perturbed space. In Figure 2(*a*)subfigure, we use t-SNE [43] to project the perturbed samples and the distortion function into 2D. The visualization reveals that distortion is localized to the samples corresponding to the perturbed class.

Zero-shot stitching We test the use of the latent functional map in the task of zero-shot stitching, as defined in [29], to combine independent encoders and decoders (e.g., classifiers, generators) without subsequent training or fine-tuning.

Experimental Setting. We consider four pre-trained image encoders (see Appendix D.2 for details) and stitch their latent spaces to perform classification using a Support Vector Machine (SVM) on two different labelings of CIFAR100 [16]: coarse and fine-grained. To evaluate the effectiveness of integrating the functional map, we extend the correspondences to determine an orthogonal transformation [22] between the latent spaces. For each encoder, we compute a graph of 3,000 points with 300 neighbors per node. We optimize the problem in Equation 1 using the first 50 eigenvectors of the graph Laplacian and consider two different descriptors: the distance functions defined from the anchors (LFM+Ortho) and the labels (LFM+Ortho (Labels)). For each dataset class, the latter provides an indicator function with 1 if the point belongs to the class and 0 otherwise. This descriptor type does not require any anchor as input, representing a pioneering example of stitching requiring no additional information beyond the dataset.
Result Analysis. Figure 3 presents the accuracy results for all possible combinations of encoder stitching. The addition of the latent functional map (LFM+Ortho) shows higher performance with a low number of anchors in both labelings of CIFAR100. Even without any anchors, the label descriptors (LFM+Ortho (Labels)) provide the best performance for the latent functional map framework in both labelings. Computing the orthogonal transformation directly from the anchors (Ortho) proves to have comparable performance only with 500 anchors, where the performance of LFM is limited by the number of eigenvectors used. This experiment shows that the latent functional map is highly effective when few anchors are available (≤ 50). It significantly enhances performance in zero-shot stitching tasks, outperforming direct orthogonal transformations at low or no anchor counts. This suggests that the latent functional map method provides a robust means of aligning latent spaces with minimal correspondence data, making it a valuable tool for tasks requiring the integration of independently trained models.

In Appendix E.2, we extend our analysis to the retrieval task, where we look for the most similar embedding in the aligned latent space. The results confirm that the latent functional map significantly enhances retrieval performance with a minimal number of anchors, making it an efficient approach for aligning latent spaces.

4. Conclusions

In this paper, we introduced latent functional maps (LFM) to enhance the understanding and utilization of neural network representations by leveraging spectral geometry for comparing and aligning different latent spaces. While LFM shows promise in unsupervised and weakly supervised settings, it faces challenges with the optimal number of eigenvectors and handling complex transformations. Future research will focus on improving scalability, effectiveness in fully unsupervised settings, and managing more complex transformations.

References

[1] Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764, 2009. ISSN 0304-3975. Algorithmic Learning Theory.

[2] Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume

162 of *Proceedings of Machine Learning Research*, pages 1607–1626. PMLR, 2022. URL https://proceedings.mlr.press/v162/barannikov22a.html.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation Learning: A Review and New Perspectives. *ArXiv preprint*, abs/1206.5538, 2012. URL https://arxiv.org/abs/1206.5538.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5: 135–146, 2017.

[5] Lisa Bonheme and Marek Grzes. How do variational autoencoders learn? insights from representational similarity. *ArXiv preprint*, abs/2205.08399, 2022. URL https://arxiv.org/abs/2205.08399.

[6] Jeff Calder and Nicolas Garcia Trillos. Improved spectral convergence rates for graph laplacians on ε-graphs and k-nn graphs. *Applied and Computational Harmonic Analysis*, 60:123–175, 2022.

[7] Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. The geometry of multilingual language model representations. 2022.

[8] MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of cka as a similarity measure in deep learning. *arXiv preprint arXiv:2210.16156*, 2022.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] James V Haxby, M Ida Gobbini, Maura L Furey, Alumit Ishai, Jennifer L Schouten, and Pietro Pietrini. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430, 2001.

[11] Judith Hermanns, Anton Tsitsulin, Marina Munkhoeva, Alex Bronstein, Davide Mottin, and Panagiotis Karras. Grasp: Graph alignment through spectral signatures. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 44–52. Springer, 2021.

[12] Harold Hotelling. Relations between two sets of variates. *Breakthroughs in statistics: methodology and distribution*, pages 162–190, 1992.

[13] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis, 2024.

[14] Max Klabunde, Tobias Schumacher, Markus Strohmaier, and Florian Lemmerich. Similarity of neural network models: A survey of functional and representational measures. *arXiv preprint arXiv:2305.06329*, 2023.

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

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

[18] Zorah Lähner and Michael 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.

[19] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H196sainb.

[20] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 991–999. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298701. URL https://doi.org/10.1109/CVPR.2015.7298701.

[21] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E. Hopcroft. Convergent learning: Do different neural networks learn the same representations? In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07543.

[22] Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. Latent space translation via semantic alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[23] Raghav Mehta, Vítor Albiero, Li Chen, Ivan Evtimov, Tamar Glaser, Zhiheng Li, and Tal Hassner. You only need a good embeddings extractor to fix spurious correlations. 2022.

[24] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, Maks Ovsjanikov, et al. Zoomout: spectral upsampling for efficient shape correspondence. *ACM TRANSACTIONS ON GRAPHICS*, 38(6):1–14, 2019.

[25] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. In *The Eleventh International Conference on Learning Representations*, 2023.

[26] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013.

[27] Mazda Moayeri, Keivan Rezaei, Maziar Sanjabi, and Soheil Feizi. Text-to-concept (and back) via cross-model alignment. In *International Conference on Machine Learning*, pages 25037–25060. PMLR, 2023.

[28] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

[29] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Locatello Francesco, Emanuele Rodola, et al. Relative representations enable zero-shot latent space communication. In *International Conference on Learning Representations*, 2023.

[30] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 360–368. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.47. URL https://doi.org/10.1109/ICCV.2017.47.

[31] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*, volume 36, pages 259–267. Wiley Online Library, 2017.

[32] Maxime Oquab, Timoth'ee Darcet, Th'eo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[33] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG)*, 31(4):1–11, 2012.

[34] Maks Ovsjanikov, Mirela Ben-Chen, Frederic Chazal, and Leonidas Guibas. Analysis and visualization of maps between shapes. In *Computer Graphics Forum*, volume 32, pages 135–145. Wiley Online Library, 2013.

[35] Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 1–60. 2016.

[36] Marco Pegoraro, Riccardo Marin, Arianna Rampini, Simone Melzi, Luca Cosmo, and Emanuele Rodolà. Spectral maps for learning on subgraphs. In *NeurIPS 2023 Workshop on Symmetry and Geometry in Neural Representations*, 2023. URL https://openreview.net/forum?id=e9JBa515z2.

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

[38] Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *International Conference on Machine Learning*, pages 9030–9039. PMLR, 2021.

[39] Gowthami Somepalli, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, and Tom Goldstein. Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective. 2022.

[40] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[41] Daniel Ting, Ling Huang, and Michael Jordan. An analysis of the convergence of graph laplacians. *arXiv preprint arXiv:1101.5435*, 2011.

[42] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Alexander M. Bronstein, Ivan V. Oseledets, and Emmanuel Müller. The shape of data: Intrinsic distance for data distributions. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=HyebplHYwB.

[43] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[44] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[45] Ivan Vulić, Sebastian Ruder, and Anders Søgaard. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.257. URL https://aclanthology.org/2020.emnlp-main.257.

[46] Fu-Dong Wang, Nan Xue, Yipeng Zhang, Gui-Song Xia, and Marcello Pelillo. A functional representation for graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[47] Liwei Wang, Lunjia Hu, Jiayuan Gu, Yue Wu, Zhiqiang Hu, Kun He, and John Hopcroft. Towards understanding learning representations: To what extent do different neural networks learn the same representation, 2018.

Appendix A. Background

This section provides the basic notions to understand the framework of functional maps applied to manifolds. We refer to [35] for a comprehensive overview.

Consider two manifolds $\mathcal{M}$ and $\mathcal{N}$ equipped with a basis such that any function $f : \mathcal{M} \to \mathbb{R}$ can be represented as a linear combination of basis functions $\mathbf{\Phi}_{\mathcal{M}}$: $f = \sum_i a_i \Phi_i^{\mathcal{M}} = \mathbf{a}\mathbf{\Phi}_{\mathcal{M}}$. Given the correspondence $T : \mathcal{M} \to \mathcal{N}$ between points on these manifolds, for any real-valued function $f : \mathcal{M} \to \mathbb{R}$, one can construct a corresponding function $g : \mathcal{N} \to \mathbb{R}$ such that $g = f \circ T^{-1}$. In other words, the correspondence T defines a mapping between two function spaces $T_F : \mathcal{F}(\mathcal{M}, \mathbb{R}) \to \mathcal{F}(\mathcal{N}, \mathbb{R})$. [33] showed how such a mapping is *linear* and can be represented as a (possibly infinite) matrix $\mathbf{C}$ such that for any function f represented as a vector of coefficients $\mathbf{a}$, we have $T_F(\mathbf{a}) = \mathbf{C}\mathbf{a}$.

The functional representation is particularly well-suited for map inference (i.e., constrained optimization). When the underlying map T (and by extension the matrix $\mathbf{C}$) is unknown, many natural constraints on the map become linear constraints in its functional representation. In practice, the simplest method for recovering an unknown functional map is to solve the following optimization problem:

$$\arg\min_{\mathbf{C}} ||\mathbf{CA} - \mathbf{B}||^2 + \rho(\mathbf{C}) \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are sets of corresponding functions expressed in the bases on $\mathcal{M}$ and $\mathcal{N}$, respectively, and $\rho(\mathbf{C})$ represents additional constraints deriving from the properties of the matrix $\mathbf{C}$ [35]. When the shapes are approximately isometric and the descriptors are well-preserved by the (unknown) map, this procedure provides a good approximation of the underlying map. In the case where the correspondence T is encoded in a matrix $\mathbf{S}$, the functional map can be retrieved as $\mathbf{C} = \mathbf{\Phi}_{\mathcal{N}}^{\dagger}\mathbf{S}\mathbf{\Phi}_{\mathcal{M}}$ where $\mathbf{\Phi}_{\mathcal{M}}$ and $\mathbf{\Phi}_{\mathcal{N}}$ are the bases of the functional spaces $\mathcal{F}(\mathcal{M}, \mathbb{R})$ and $\mathcal{F}(\mathcal{N}, \mathbb{R})$, respectively, and † denotes the pseudo-inverse.

Appendix B. Related Work

Similarity between latent spaces Comparing representations learned by neural models is of fundamental importance for a diversity of tasks, ranging from representation analysis to latent space alignment and neural dynamics. In order to do so, a similarity measure between different spaces must be defined [14]. This can range from functional similarity (matching the performance of two models) to similarity defined in representational space [15], which is where our framework falls in. A classical statistical method is Canonical Correlation Analysis (CCA) [12], known for its invariance to linear transformations. Various adaptations of CCA aim to enhance robustness, such as through Singular Value Decomposition (SVD) and Singular Vector Canonical Correlation Analysis (SVCCA) [37], or to decrease sensitivity to perturbations using methods like Projection-Weighted Canonical Correlation Analysis (PWCCA) [28]. Closely related to these approaches, Centered Kernel Alignment (CKA) [15] measures the similarity between latent spaces while ignoring orthogonal transformations. However, recent research [8] reveals that CKA is sensitive to shifts in the latent space.

We propose to leverage LFMs as a tool to measure the similarity, or how much two spaces differ from an isometry w.r.t. to the metric that has been used to construct the graph.

Latent communication This relatively new concept, introduced by [29], builds on the hypothesis that latent spaces across neural networks (pre-)trained with many variation factors, from random seed initialization to architecture or even data modality, are intrinsically compatible. This notion is supported by numerous empirical studies [2, 3, 5, 7, 15, 19–21, 28, 30, 42, 45], with the phenomenon being particularly evident in large and wide models [23, 39]. The core idea is that relations between data points (i.e., distances according to some metric) are preserved across different spaces because the high-level semantics of the data are the same and neural networks learn to encode them similarly [13] . With this "relative representation", the authors show that it is possible to *stitch* [20] together model components coming from different models, with little to no additional training as long as a partial correspondence of the spaces involved is known.

Indeed, [18, 22, 25, 27] show that a simple linear transformation is usually enough to map one latent space into another measured by performance on desired downstream tasks.

With LFMs, we change the perspective from merely relating samples of distinct latent spaces to relating function spaces defined on the manifold that the samples approximate, showing that processing information in this dual space is convenient as it boosts performance while also being interpretable.

Functional Maps. The representation we propose is directly derived from the functional maps framework for smooth manifolds introduced in the seminal work by [33]. This pioneering study proposed a compact and easily manipulable mapping between 3D shapes. Subsequent research has aimed at enhancing this framework. For instance, [31] introduced regularization techniques to improve the informativeness of the maps, while [24] developed refinement methods to achieve more accurate mappings. The functional map framework has been extended as well outside the 3d domain, for example, in [46] and [11], who applied the functional framework to model correspondences between graphs, and in [36], who demonstrated its utility in graph learning tasks. In particular, they have shown that the functional map representation retains its advantageous properties even when the Laplace basis is computed on a graph.

Inspired by these advancements, our work leverages the functional representation of latent spaces. We demonstrate how this representation can be easily manipulated to highlight similarities and facilitate the transfer of information between different spaces, thereby extending the applicability of the functional maps framework to the domain of neural latent spaces.

Appendix C. Latent Functional Map

C.1. Details

C.1.1. BUILDING THE GRAPH

To leverage the geometry of the underlying manifold, we model the latent space of a neural network building a symmetric k-nearest neighbor (k-NN) graph [1]. Given a set of samples $X = \{x_1, \dots, x_n\}$, we construct an undirected weighted graph $G = (X, E, \mathbf{W})$ with nodes X, edges E, and weight matrix $\mathbf{W}$. The weight matrix is totally characterized by the choice of distance function $d(x_i, x_j)$ with $x_i, x_j \in X$. Suitable choices include the L2 metric or the angular distance. Edges E are defined as $E = \{(x_i, x_j) \in X \times X \mid x_i \sim_{\text{k}} x_j \text{ or } x_j \sim_{\text{k}} x_i\}$, where $x_i \sim_{\text{k}} x_j$ indicates that x_j is among the k nearest neighbors of x_i. The weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}_{\geq 0}$ assigns a weight $\omega(x_i, x_j)$ to each edge $(x_i, x_j) \in E$, and $\mathbf{W}(i, j) = 0$ otherwise.

Next, we define the associated weighted graph Laplacian $\mathcal{L}_G = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is the diagonal degree matrix with entries $\mathbf{D}(i, i) = \sum_{j=1}^{n} \mathbf{W}(i, j)$. $\mathcal{L}_G$ is a positive semi-definite, self-adjoint operator [44]), therefore, it admits an eigendecomposition $\mathcal{L}_G = \mathbf{\Phi}_G \mathbf{\Lambda}_G \mathbf{\Phi}_G^T$, where $\mathbf{\Lambda}_G$ is a diagonal matrix containing the eigenvalues, and $\mathbf{\Phi}_G$ is a matrix whose columns are the corresponding eigenvectors. The eigenvectors form an orthonormal basis for the space of functions defined on the graph nodes (i.e., $\mathbf{\Phi}_G^T \mathbf{\Phi}_G = \mathbf{I}$).

Throughout this paper, we assume the eigenvalues (and corresponding eigenvectors) are sorted in non-descending order $0 = \Lambda_1 \leq \Lambda_2 \leq \cdots \leq \Lambda_n$. One may consider a subset of eigenvectors, namely those associated with the k smallest eigenvalues, to compactly approximate a graph signal, employing techniques akin to Fourier analysis.

As demonstrated in many recent works [6, 41], the eigenvalues and eigenvectors of the graph Laplacian associated with a k-NN graph approximate the weighted Laplace-Beltrami operator, placing us in a setting similar to the original one of [33].

C.1.2. SPACE OF FUNCTIONAL COEFFICIENTS

The space of functional coefficients offers an alternative representation for points in the latent space $\mathcal{X}$. Using the equation $\hat{f}_G = \mathbf{\Phi}_G^T f_G$, any function $f_G \in \mathcal{F}(G, \mathbb{R})$ can be uniquely represented by its functional coefficients $\hat{f}_G$. We leverage this property to represent any point $x \in \mathcal{X}$ as a distance function $f_d \in \mathcal{F}(G, \mathbb{R})$ from the set of points X_G, which correspond to the nodes of the graph G. The functional map $\mathbf{C}$ between two latent spaces $\mathcal{X}$ and $\mathcal{Y}$ aligns their functional representations, enabling the transfer of any function from the first space to the second. This functional alignment can be used similarly to the method proposed by [29] to establish a "relative" space where the representational spaces $\mathcal{X}$ and $\mathcal{Y}$ are aligned.

C.1.3. EXTENDING SPACES CORRESPONDENCES

As explained in Section A, the functional map $\mathbf{C}$ represents the bijection T in a functional form. [33] demonstrated that this bijection can be retrieved as a point-to-point map by finding the nearest neighbor for each row of $\mathbf{\Phi}_{G_Y}\mathbf{C}$ in $\mathbf{\Phi}_{G_X}$. This process can be efficiently implemented using algorithms such as kd-tree. Given a few correspondences (anchors) between the two spaces $\mathcal{X}$ and $\mathcal{Y}$, we can extend these correspondences to the entire set of nodes X and Y. This extended set of anchors can then be used to determine a transformation between the latent spaces, as described by [22]. In the following section, we demonstrate that by using a small number of anchors (≤ 50), we can retrieve optimal transformations that facilitate near-perfect stitching and retrieval.

C.2. Additional Regularizers

In Equation 1, we improve the computation of the functional map by incorporating two additional regularizers: Laplacian commutativity and descriptor operator commutativity. Both regularizers exploit the preservation of linear functional operators $\mathbf{S}^G : \mathcal{F}(G, \mathbb{R}) \to \mathcal{F}(G, \mathbb{R})$, enforcing that the functional map $\mathbf{C}$ commutes with these operators: $||\mathbf{S}_i^G\mathbf{C} - \mathbf{C}\mathbf{S}_i^{G_X}|| = 0$.

The Laplacian commutativity regularizer, first introduced by [33], is formulated as:

$$\rho_{\mathcal{L}}(\mathbf{C}) = ||\mathbf{\Lambda}_{G_Y}\mathbf{C} - \mathbf{C}\mathbf{\Lambda}_{G_X}||^2 \tag{3}$$

where $\mathbf{\Lambda}_G$ represents the diagonal matrices of eigenvalues. This regularizer ensures that the functional map $\mathbf{C}$ preserves the spectral properties of the Laplacian.

The descriptor operator commutativity regularizer, introduced by [31], extracts more detailed information from a given descriptor, resulting in a more accurate functional map even with fewer descriptors. The formulation of this regularizer is as follows:

$$\rho_f(\mathbf{C}) = \sum_i ||\mathbf{S}_i^{G_Y}\mathbf{C} - \mathbf{C}\mathbf{S}_i^{G_X}||^2 \tag{4}$$

where $\mathbf{S}_i^G = \mathbf{\Phi}_G^T \text{Diag}(f_i^G)\mathbf{\Phi}_G$ are the descriptor operators.

Appendix D. Experimental details

D.1. Architecture Details

All non-ResNet architectures are based on All-CNN-C [40]

Tiny-10
3×3 conv. 16-BN-ReLu $\times 2$
3×3 conv. 32 stride 2-BN-ReLu
3×3 conv. 32-BN-ReLu $\times 2$
3×3 conv. 64 stride 2-BN-ReLu
3×3 conv. 64 valid padding-BN-ReLu
1×1 conv. 64-BN-ReLu
Global average pooling
Logits

Table 1

D.2. Pre-trained models

In Section 3 we used four pretrained models: 3 variations of [9] ('google-vit-base-patch16-224', 'google-vit-large-patch16-224', 'WinKawaks-vit-small-patch16-224') and the model proposed by [32] ('facebook-dinov2-base').

D.3. Parameters and resources

In all our experiments we used gpu rtx 3080ti and 3090. In order to compute the eigenvector and functional map on a graph of 3k nodes we employ not more than 2 minutes.

D.4. Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank (MRR) is a commonly used metric to evaluate the performance of retrieval systems [29]. It measures the effectiveness of a system by calculating the rank of the first relevant item in the search results for each query.

To compute MRR, we consider the following steps:

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

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{r_i} \tag{5}$$

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.

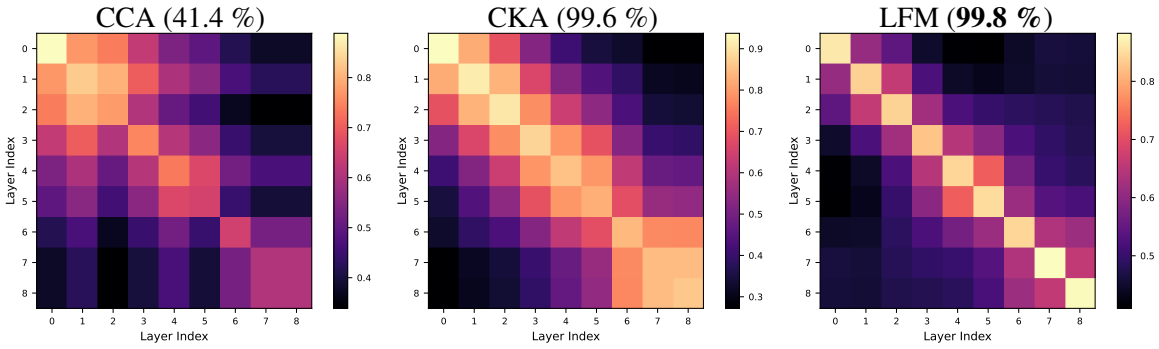


Figure 4: **Similarity across layers** Similarity matrices between internal layer representations of `CIFAR10` comparing our LFM-based similarity with the CCA and CKA baselines, averaged across 10 models. For each method, we report the accuracy scores for matching the corresponding layer by maximal similarity.

Appendix E. Additional Results

E.1. Analysis

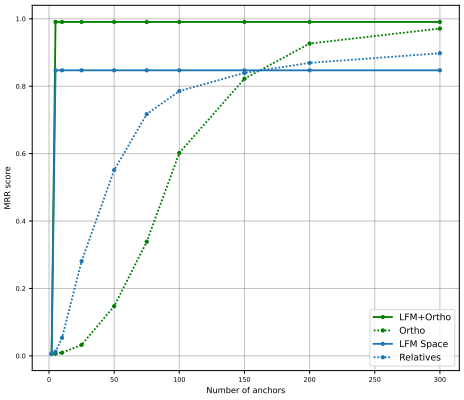
Experimental setting: In order to validate experimentally if LFMs can serve as a good measure of similarity between distinct representational spaces, we run the same sanity check as in [15]. We train 10 CNN models (the architecture is depicted in Appendix D.1) on the `CIFAR-10` dataset [16], changing the initialization seed. We compare their inner representations at each layer, excluding the logits and plot them as a similarity matrix, comparing with Central Kernel Alignment (CKA) measure [15] and Canonical Correlation Analysis (CCA) [12, 37]. We then measure the accuracy of identifying corresponding layers across models and report the results comparing with CKA and CCA as baselines. For CCA, we apply average pooling on the spatial dimensions to the embeddings of the internal layers, making it more stable numerically and boosting the results for this baseline compared to what was observed in [15].

Result analysis Figure 4 shows that our LFM-based similarity measure behaves correctly as CKA does. Furthermore, the similarities are less spread around the diagonal, favoring a slightly higher accuracy score in identifying the corresponding layers across models.

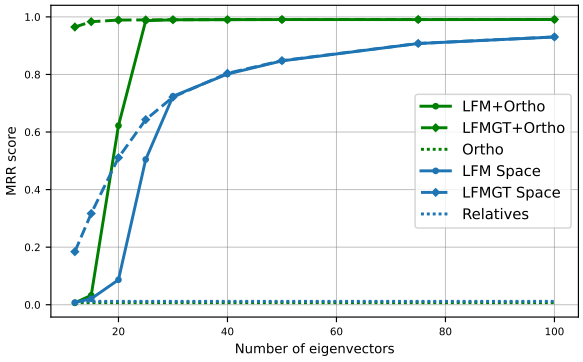
E.2. Retrieval

We extend our analysis to the retrieval task, where we look for the most similar embedding in the aligned latent space.

Experimental Setting We consider two different English word embeddings, FastText [4] and Word2Vec [26]. Following the approach of [29], we extract embeddings of 20K words from their shared vocabulary using pre-trained models. We use 2K random corresponding samples to construct the k-NN graphs and evaluate the retrieval performance on the remaining 18K word embeddings. We test two settings in our experiments: (i) Aligning functional coefficients (LFM Space). (ii) Computing an orthogonal transformation using the correspondences obtained by the functional map (LFM+Ortho). For this experiment, we construct k-NN graphs with a neighborhood size of 300 and compute the functional map using the first 50 eigenvectors. We evaluate the methods' performance

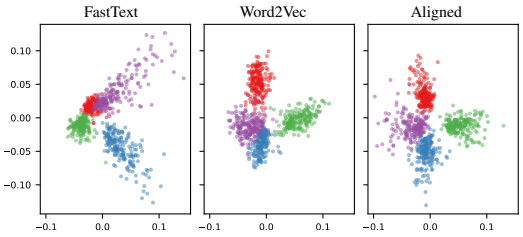


(*a*) MRR score at increasing number of anchors

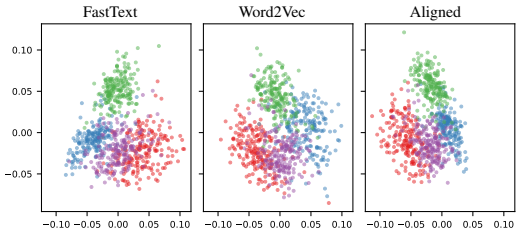


(*b*) MRR score at increasing number of eigenvectors

Figure 5: **Ablation on Retrieval of word embeddings.** We compare the retrieval performance of the functional map framework with state-of-the-art models as the number of anchors increases. The left panel shows the Mean Reciprocal Rank (MRR) across different numbers of anchors. The right panels depict the first two components of PCA for a subsample of the latent space (b) and the functional space (c), both before and after alignment using the functional map.



(*a*) Latent space



(*b*) Functional space

Figure 6: **Retrieval of word embeddings.** The panels depict the first two components of PCA for a subsample of the latent space (b) and the functional space (c), both before and after alignment using the functional map.

Table 2: **MRR Score for the retrieval of word embeddings.** We report the value of the results depicted in Figure 6 adding more kind transformation between spaces (Orthogonal, Linear and Affine).

	Number of anchors									
Method	2	5	10	25	50	75	100	150	200	300
LFM+Ortho	0.01	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
LFM+Linear	0.01	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
LFM+Affine	0.01	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Ortho	0.01	0.01	0.01	0.03	0.15	0.34	0.60	0.82	0.93	0.97
Linear	0.01	0.01	0.01	0.05	0.26	0.49	0.66	0.77	0.74	0.01
Affine	0.01	0.01	0.01	0.04	0.19	0.45	0.64	0.81	0.89	0.95
LFM Space	0.01	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
Relatives	0.01	0.01	0.05	0.28	0.55	0.72	0.79	0.84	0.87	0.90

using the Mean Reciprocal Rank (MRR), as detailed in Appendix D.4. Our functional map methods are compared with the method proposed by [29] (Relatives) and the orthogonal transformation method proposed by [22] (Ortho).

Result Analysis Figure 6 shows the performance of these methods as the number of anchors increases. The numerical results are detailed in Table 2. The functional map significantly improves performance with just 5 anchors, achieving an MRR of over 0.8. As the number of anchors increases, the performance of competing methods improves but still falls short of FMAP+Transform at 300 anchors, which reaches an MRR of 0.99. Interestingly, the performance of the functional map methods does not improve beyond 5 anchors, suggesting that this number of anchors is sufficient to achieve an optimal functional map between the spaces. In Table 2, we report the numerical results for the experiment in Figure 6 adding more transformations from the method of [22]: orthogonal (Ortho), linear (Linear) and affine (Affine). From the value in the table, we can see that all the methods that involve the latent functional map (LFM) saturate at 5 anchors, reaching top performance. We further analyze how the results improve as the number of eigenvectors used to compute the functional map increases. In Figure 5(*b*)subfigure, we show how the performance of the latent functional map methods depends on the number of eigenvectors used to compute the map. In particular, we notice that the performance drastically increases at 25 eigenvectors, reaching the same score when using the functional map computed from the ground truth correspondences (LFMGT). These results confirm that the latent functional map is a valuable tool in settings with little knowledge about correspondences. It significantly enhances retrieval performance with a minimal number of anchors, making it an efficient approach for aligning latent spaces. Moreover, its performance can be improved using a higher number of eigenvectors.