NAM: Neural Adjoint Maps for refining shape correspondences

GIULIO VIGANÒ, University of Milano-Bicocca, Italy
MAKS OVSJANIKOV, LIX, Ecole Polytechnique, France
SIMONE MELZI, University of Milano-Bicocca, Italy

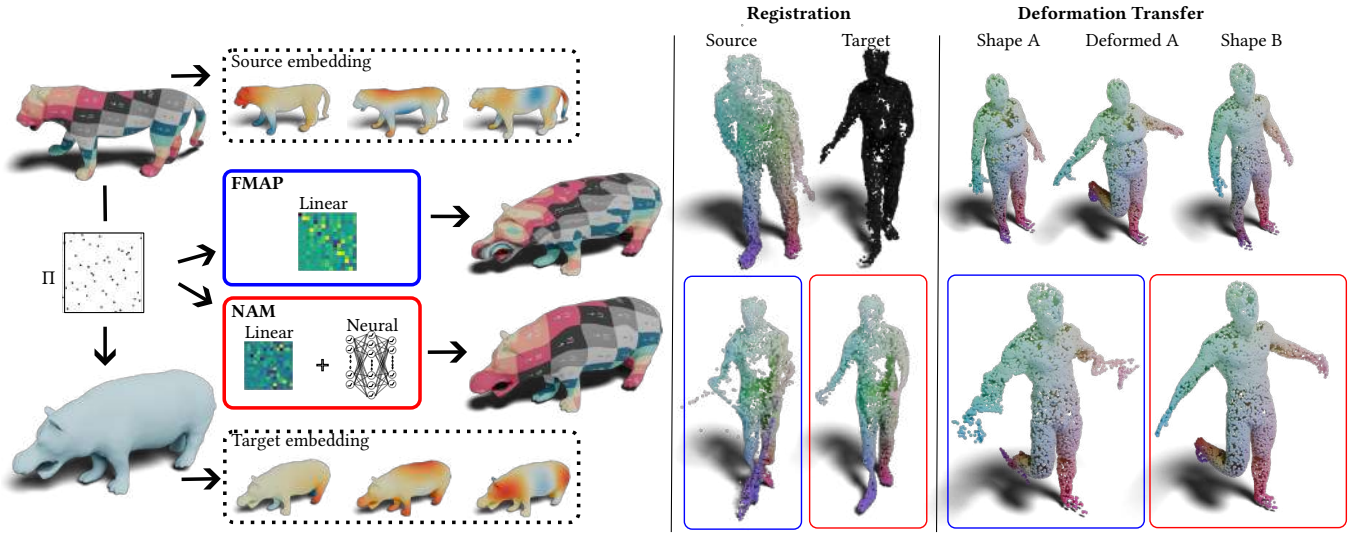

Fig. 1. A visualization of the proposed approach (NAM), a generalization of the functional maps framework (FMAP). On the left, given a pair of shapes equipped with an embedding (dotted boxes) and an unknown correspondence Π between them, we estimate an FMAP representation of Π as a linear operator (blue), and our neural map representation (red). From these representations, we estimate Π depicted as texture transfer. On the right, we compare FMAP (blue) and NAM (red) in challenging applications on point clouds: registration and deformation transfer. NAM outperforms its linear counterpart in all cases.

In this paper, we propose a novel approach to refine 3D shape correspondences by leveraging multi-layer perceptions within the framework of functional maps. Central to our contribution is the concept of *Neural Adjoint Maps*, a novel neural representation that generalizes the traditional solution of functional maps for estimating correspondence between manifolds. Fostering our neural representation, we propose an iterative algorithm explicitly designed to enhance the precision and robustness of shape correspondence across diverse modalities such as meshes and point clouds. By harnessing the expressive power of non-linear solutions, our method captures intricate geometric details and feature correspondences that conventional linear approaches often overlook. Extensive evaluations on standard benchmarks and challenging datasets demonstrate that our approach achieves state-of-the-art accuracy for both isometric and non-isometric meshes and for point clouds where traditional methods frequently struggle. Moreover, we show the versatility of our method in tasks such as signal and neural field transfer, highlighting its broad applicability to domains including computer graphics, medical imaging, and other fields demanding precise transfer of information among 3D shapes. Our work sets a new standard for shape correspondence refinement, offering robust tools across various applications.

CCS Concepts: • **Computing methodologies** → **Shape analysis**; **Machine learning**.

Additional Key Words and Phrases: Shape Matching, Point Cloud, Machine Learning

ACM Reference Format:
Giulio Viganò, Maks Ovsjanikov, and Simone Melzi. 2025. NAM: Neural Adjoint Maps for refining shape correspondences. *ACM Trans. Graph.* 44, 4, Article 60 (August 2025), 15 pages. https://doi.org/10.1145/3730943

Authors' Contact Information: Giulio Viganò, University of Milano-Bicocca, Italy, g.vigan36@campus.unimib.it; Maks Ovsjanikov, movsjanka@gmail.com, LIX, Ecole Polytechnique, France; Simone Melzi, simone.melzi@unimib.it, University of Milano-Bicocca, Italy.

ACM 1557-7368/2025/8-ART60
https://doi.org/10.1145/3730943

1 INTRODUCTION

Shape correspondence or shape matching is a fundamental task in shape analysis, involving the estimation of pointwise correspondences between discrete 3D shapes. This task has broad applications, including statistical shape analysis [Bogo et al. 2014], medical imaging [Magnet et al. 2023], and deformation transfer [Sumner and Popović 2004], making it a key focus in recent research [Deng et al. 2022; Sahillioğlu 2020]. In the discrete setting, shape correspondence consists of estimating a permutation between two shapes encoded as a set of vertices (for meshes) or points (for point clouds) such that this permutation respects some geometric or semantic similarity between the shapes.

Despite recent developments, the problem of shape correspondence remains a substantial challenge, particularly in complex settings with strong variability between shapes. State-of-the-art methods often involve training procedures to learn features representing similarities between shapes, which commonly achieve good results. However, despite their efficacy, they can fail if applied to shapes that are too different from those used during training. In this scenario, a robust method to refine correspondences at test time, which should be independent of discretization and generalize well to shape variations never seen at training time, becomes crucial.

We can broadly classify existing refinement solutions into intrinsic and extrinsic approaches. Intrinsic methods, such as those based on the Functional Map [Ovsjanikov et al. 2012] framework, as [Melzi et al. 2019b], assume isometry and can struggle with non-isometric deformations. Conversely, extrinsic methods, such as [Li 2022], exploit constraints based on the 3D embedding of the shapes, which are effective but computationally expensive and can struggle with strong variations and topological noise.

Our proposed solution arises exactly in this context by addressing these limitations and being effective in the different scenarios, handling both intrinsic and extrinsic challenges. From a theoretical perspective, our method introduces a novel notion of Neural Adjoint Map (NAM), a non-linear generalization of the functional maps framework. Functional maps encodes correspondences as linear operators between functional spaces defined on the shapes by selecting appropriate functional bases. These bases effectively serve as high-dimensional embeddings of the points and enable correspondence estimation by aligning these embeddings. NAM replaces the linear operator and extends to more general embedding in the functional maps procedure, retaining the ability to handle isometric pairs and extending the framework to non-isometric and challenging cases. With NAM, we propose a novel correspondence representation that leverages the computation of neural function between high-dimensional embeddings. Combining linear and non-linear modules, NAM encodes the correspondence as a smooth and non-linear map, which acts like a functional map operator. Fostering the NAM representation, we develop a refinement strategy to compute accurate correspondences between shapes across different modalities. We demonstrate the efficacy and applicability of our refinement procedure across several scenarios, including isometric meshes, non-isometric meshes, and point clouds. Moreover, we test the capability of our method to exploit different high-dimensional embeddings, showing remarkable flexibility.

To summarize, our contributions are:

- We define a novel non-linear representation for correspondences inspired by the functional maps framework, but that solves some of its limitations.
- We develop a refinement strategy to improve the accuracy of estimated correspondences between shapes across different modalities.
- We evaluate the proposed technique on different datasets by comparing it with existing solutions.
- We test the refined correspondence by adopting them to target challenging applications such as point cloud registration and deformation transfer.

Many shape correspondence pipelines lack a unified method that effectively refines correspondences, handling different modalities, representations, and challenging scenarios. Our work proposes to fill this gap in the context of refining correspondences, providing a robust and versatile solution for refining shape correspondences.

2 RELATED WORKS

2.1 Shape matching with Functional Maps

A well-studied task for computer graphics applications is shape matching, whose goal is to find correspondences between the elements (points or vertices) of two discrete shapes, such as point clouds or meshes.

In the case of meshes, Functional Maps framework[Ovsjanikov et al. 2012] has achieved impressive results both exploiting optimization procedures [Maggioli et al. 2025; Nogneng et al. 2018; Nogneng and Ovsjanikov 2017; Ovsjanikov et al. 2016; Ren et al. 2019, 2018; Rodolà et al. 2017] and more recent deep learning techniques [Cao and Bernard 2023; Donati et al. 2022; Eisenberger et al. 2020; Litany et al. 2017; Sun et al. 2023]. These methods exploit a functional representation of the correspondence, which can be accurately estimated on meshes and is particularly effective for isometric shapes. Recent learning-based alternatives leverage this notion only at training time as a regularizer to learn universal features [Cao et al. 2023; Sun et al. 2023] from which they directly estimate the correspondence at inference time.

However, functional maps and related techniques do not achieve the same accuracy for non-isometric meshes. This performance gap is due to the difference in the adopted functional representation of the shapes when these are non-isometric. Furthermore, in dealing with point clouds, the functional approach is not so effective because, in this setting, the surface information is less accurate and the computation of the functional representation can be unstable between shapes. For these reasons, follow-up solutions tried to deal with these issues by proposing novel representations for the functional space [Azencot et al. 2019; Hartwig et al. 2023; Kovnatsky et al. 2013; Melzi 2019; Melzi et al. 2020, 2018] or learn more stable representation for point clouds [Marin et al. 2020; Viganò and Melzi 2024]. An alternative approach is to learn different embeddings based on reconstruction loss and solve for the correspondence using these embeddings as features and without involving any functional representation [Deng et al. 2023; Ginzburg and Raviv 2021]. Unfortunately, these representations inevitably require tuning of parameters to target different deformations or create a bias towards some other category of shapes. Other recent approaches involve considering attention mechanisms in the learning phase [Li and Harada 2022; Raganato et al. 2023; Riva et al. 2024], also in the functional setting [Li et al. 2022].

2.2 Map refinement

Despite the effectiveness of the state-of-the-art approaches, obtaining a high-quality correspondence on any input pairs is still far, even if necessary for real applications. For this reason, a line of research focuses on designing a procedure to refine a given initial correspondence. A good refinement should improve the quality of any input correspondence as efficiently as possible.

With the functional maps paradigm, intrinsic approaches refine the maps iteratively by converting between pointwise and functional representation [Magnet et al. 2022; Melzi et al. 2019b; Ren et al. 2020, 2021]. These methods are efficient and robust to variations in the discretization, but often fail in the case of non-isometric pairs. In [Hartwig et al. 2023], the ZoomOut algorithm [Melzi et al. 2019b] has been extended to perform upsampling of elastic basis, and the same is done in [Viganò and Melzi 2024] for learned embeddings [Marin et al. 2020]. These refinements work well but are limited to pairs with similar discretizations and are too dependent on the unstable quality of the representation.

An alternative refinement approach is to update the correspondence by optimizing for an extrinsic deformation of the shapes. These methods often define some encoding of the deformations and compute explicit constraints such as ARAP energy [Sorkine and Alexa 2007] or the Chamfer distance. These methods can achieve accurate correspondences, but they are typically expensive. Recently, NDP [Li 2022] and Nerfies [Park et al. 2021] optimize extrinsic deformation by applying neural fields. These methods can realize good registrations if the deformed shapes are near-rigidly alignable, but they suffer in the case of pose variation or strong deformations. In [Cao et al. 2023], the authors propose a method to improve learned features at test time by minimizing Dirichlet energy. However, these methods work only for a specific type of initialization; thus, they cannot refine any input correspondence.

Instead, with NAM, we propose a method that can be applied both on non-isometric meshes and point clouds, with any embedding representing the shape, and can refine any input correspondences, without the need of previous training. Table 1 summarises the properties of different refinement methods.

Table 1. Comparison of the properties of different refinement methods. Intrinsic indicates that the method leverages intrinsic information, so it's more stable under strong pose variations. Non-Isometric and Point Cloud indicate the kind of pairs a method can deal with. Fast means that the refinement method has a computational time shorter than the average time for estimating a correspondence. Agnostic indicates that the refinement method can improve any input correspondence. Here are the reference for each method: ZO [Melzi et al. 2019b], Elastic ZO [Hartwig et al. 2023], NDP [Li 2022], ULRSM[Cao et al. 2023].

Method	Intrinsic	Non Isometric	Point Cloud	Fast	Agnostic
ZO	✓	✗	✓	✓	✓
Elastic ZO	✗	✓	✗	✓	✓
NDP	✗	✗	✓	✓	✓
ULRSM	✓	✓	✗	✗	✗
NAM	✓	✓	✓	✓	✓

2.3 Neural Deformation Fields and Neural Prior

Neural fields, a tool of increasing interest in computer graphics, recently met the shape correspondence task. In particular, most of these approaches exploit neural fields to represent the non-rigid transformation between the 3D extrinsic embeddings of the shapes. Several works optimize for displacement fields between 3D coordinates [Corona et al. 2022; Marin et al. 2024; Sundararaman et al. 2022], while NDP [Li 2022] and Nerfies [Park et al. 2021] optimize for a warp field on a Fourier frequency bandwidth. Other works [Aigerman et al. 2022; Sundararaman et al. 2024] adopt the representation of deformations from [Sumner and Popović 2004] to learn discretization agnostic fields. These methods achieve good results, but they still rely on a mesh structure, and extending them to point clouds is not straightforward.

In recent years, various works have leveraged the properties of the so-called *neural-prior* to constrain optimization problems by exploiting untrained neural networks. [Ulyanov et al. 2020], which adopts the neural prior to deblur images, is the first work in this direction. Follow-up works extend this idea to rigid registration [Li et al. 2021], handle-guided shape-deformation [Tang et al. 2023], and correspondence learning [Attaiki and Ovsjanikov 2023].

3 BACKGROUND AND NOTATION

3D shapes. We model a 3D shape $\mathcal{X}_1$ as a compact two-dimensional manifold embedded in $\mathbb{R}^3$. Along with the surface, we will consider the space of squared-integrable real-valued functions defined on the surface of the shape $\mathcal{L}^2(\mathcal{X}_1)$, defined as

$$\mathcal{L}^2(\mathcal{X}_1) := \{f : \mathcal{X}_1 \to \mathbb{R}, s.t. \int_{\mathcal{X}_1} |f(x)|^2 dx < \infty\}.$$

This functional space is known to be a Hilbert space with the inner product

$$\langle f, g \rangle_{\mathcal{L}^2(\mathcal{X}_1)} = \int_{\mathcal{X}_1} f(x)g(x)dx. \tag{1}$$

From a computational point of view, we can discretize the shape in different ways, such as point clouds and meshes, to name two of the most adopted ones. These discrete representations usually involve an unordered collection of $n_1 \in \mathbb{N}$ points $\{x_i\}_{i=0}^{n_1}$, represented by their 3D coordinates. In the discrete setting, each function $f : \mathcal{X}_1 \to \mathbb{R}$ is represented as a vector $f \in \mathbb{R}^{n_1}$, where each entry $f_i = f(x_i), \forall i \in \{1, \dots, n_1\}$. For each shape $\mathcal{X}_1$, we can consider the mass matrix $M_1 \in \mathbb{R}^{n_1 \times n_1}$, a diagonal matrix with entries equal to the mass values m_i of each point x_i of the discretized shape. With this matrix, we can discretize the inner product in Equation (1) as $\langle f, g \rangle_{\mathcal{L}^2(\mathcal{X}_1)} = f^\top M_1 g$. Another important operator is the Laplace Beltrami Operator (LBO) [Levy 2006] $\Delta_1 : \mathcal{L}^2(\mathcal{X}_1) \to \mathcal{L}^2(\mathcal{X}_1)$, that we can discretize as a matrix in $\mathbb{R}^{n_1 \times n_1}$ [Pinkall and Polthier 1993]. By computing the eigendecomposition of this operator, we obtain its non-negative eigenvalues $\{\lambda_1^i\}_{i=0}^{n_1} \in \mathbb{R}$ and associated eigenfunctions $\{\phi_1^i\}_{i=0}^{n_1} \in \mathcal{L}^2(\mathcal{X}_1)$ that form an orthonormal basis (the LBO basis) for $\mathcal{L}^2(\mathcal{X}_1)$. In the discrete setting, the number of these eigenfunctions is exactly n_1. For efficiency reasons, it is common practice to use only a truncated subset of the eigenfunctions associated with the $k \leq n_1$ smallest eigenvalues to provide a compact low-pass approximation of the functions in a Fourier-like representation. We can store this truncated basis in a matrix $\Phi_1^k = [\phi_1^1, \dots, \phi_1^k] \in \mathbb{R}^{n_1 \times k}$.

Shape matching. If we consider two 3D shapes $\mathcal{X}_1 = \{x_i\}_{i=1}^{n_1}$ and $\mathcal{X}_2 = \{y_j\}_{j=1}^{n_2}$, we can consider a point-to-point correspondence T_{12} which maps each point x_i into its corresponding point y_j. The goal of shape matching is to estimate this correspondence. T_{12} can be represented as the matrix $\Pi_{21} \in \mathbb{R}^{n_1 \times n_2}$ such that $\Pi_{21}(i, j) = 1$, if $T_{12}(x_i) = y_j$ and $\Pi_{21}(i, j) = 0$ otherwise. If the correspondence is bijective, Π_{21} is a permutation matrix.

3.1 Functional Maps

A correspondence $T_{12} : \mathcal{X}_1 \to \mathcal{X}_2$ induces via pull-back a linear map $T_{21}^F : \mathcal{L}^2(\mathcal{X}_2) \to \mathcal{L}^2(\mathcal{X}_1)$, defined as $T_{21}^F(g) = g \circ T_{12} \in \mathcal{L}^2(\mathcal{X}_1)$, $\forall g \in \mathcal{L}^2(\mathcal{X}_2)$. If we equip the functional spaces with bases Φ_1, Φ_2, we can represent each function as a vector of the coefficients of its representation in the basis. Therefore, the functional map T_{21}^F can be encoded as a matrix C_{21} that maps coefficients. For details on this derivation, we refer to the supplementary.

Given Π_{21}, the functional map can be explicitly computed as

$$C_{21} = \Phi_1^\dagger \Pi_{21} \Phi_2, \tag{2}$$

where † denotes the Moore–Penrose pseudoinverse. This definition holds for any basis even if the standard choice is the LBO basis.

Adjoint Operator. The functional map T_{21}^F univocally induces the *adjoint operator* [Hartwig et al. 2023; Huang and Ovsjanikov 2017; Pai et al. 2021] as the unique operator $T_{12}^A : \mathcal{L}^2(\mathcal{X}_1) \to \mathcal{L}^2(\mathcal{X}_2)$, that satisfies $\langle T_{12}^A f\ ,\ g \rangle_{\mathcal{L}^2(\mathcal{X}_2)} = \langle f\ ,\ T_{21}^F g \rangle_{\mathcal{L}^2(\mathcal{X}_1)}$.

In the discrete setting, the adjoint operator T_{12}^A corresponds to a matrix $\Gamma_{12} \in \mathbb{R}^{n_1 \times n_2}$ that we can compute as $\Gamma_{12} = M_2^{-1} \Pi_{21}^\top M_1$ [Pai et al. 2021]. In the case of isometric shapes with the same connectivity $\Gamma_{12} = \Pi_{21}^\top = \Pi_{12}$.

Equipping the functional spaces with bases, as done in [Hartwig et al. 2023; Pai et al. 2021], we can represent the adjoint operator as the matrix:

$$A_{12} = \widetilde{M}_2^{-1} C_{21}^\top \widetilde{M}_1, \tag{3}$$

where $\widetilde{M}_i = \Phi_i^\top M_i \Phi_i$. In the case of orthonormal bases, as with LBO, $A_{12} = C_{21}^\top$. We refer to [Hartwig et al. 2023; Pai et al. 2021] for further details.

Map conversion. The functional maps framework and related shape matching solutions leverage functional maps to represent the correspondence with a linear operator. On the full basis, the correspondence can be recovered from the functional map. Indeed, each point $x \in \mathcal{X}_1$ can be represented through δ_x, the Dirac delta distribution centered in x, defined as the unique distribution for which holds that $\langle \delta_x, f \rangle = f(x)$, $\forall f \in \mathcal{L}^2(\mathcal{X}_1)$. By selecting a basis for the functional spaces, a set of coefficients can encode this distribution for every point on the two shapes in the corresponding coefficients. For the LBO basis for instance, $\Phi_1(x)$, the row corresponding to x in the matrix Φ_1, are the coefficients of δ_x. A common practice is to leverage functional operators to transfer deltas from one space to the other and recover the correspondence as the solution of the nearest search (denoted NS and applied on rows for matrices) between the spectral representations of deltas.

We can summarize the most related and impactful recovering strategies as:

Functional maps [Ovsjanikov et al. 2012]: $T_{12} = NS\Big(\Phi_1, \Phi_2 C_{21}^\top\Big)$. (4)

Adjoint operator [Pai et al. 2021]: $T_{12} = NS\Big(\Phi_1 A_{12}^\top, \Phi_2\Big)$. (5)

Elastic [Hartwig et al. 2023]: (6)

$$T_{12} = NS\Big((\sqrt{\widetilde{M}_1}\Phi_1^\dagger M_1^{-1})^\top, (\sqrt{\widetilde{M}_1} C_{21} \Phi_2^\dagger M_2^{-1})^\top\Big). \tag{7}$$

In [Hartwig et al. 2023], the authors derive (7) to generalize the functional maps framework to non-orthonormal bases. $(\Phi_1^\dagger M_1^{-1})^\top$ are indeed the coefficients of the Dirac delta distributions centered at the points of $\mathcal{X}_1$ for a generic basis Φ_1. In the case of an orthonormal basis, (4) and (7) are equivalent. In [Pai et al. 2021], the adjoint operator has been proven to be the operator that maps delta functions, and so equation (5) is theoretically the most correct one, however in practice, all these methods introduce approximations.

Truncated basis. Indeed, the above considerations are based on the assumption of having a full basis, which represents the entire functional space. But, functional map efficacy is based on adopting a truncated basis with $k << n_1, n_2$. In this way, we can optimize for a compact functional map that acts as a low-pass filter, helping to extract and optimize global map properties.

However, this introduces strong approximations in delta function representation and in the point-to-point correspondences that can be recovered from compact linear maps. Indeed, if we refer to the row corresponding to x in the matrix Φ_1^k as the *spectral embedding* of dimension k of x, the truncation induces a k dimensional embedding of the shapes, which cannot be completely aligned by linear operators without imposing some prior on the embeddings themselves, as we will prove in Section 4.2.

Our work tackle this limitation, proposing a new representation that can represent a broader space of correspondences.

3.2 Refinement

To keep the advantages of optimizing compact linear functional maps and overcome the limitations of having strong approximations of shape representations, different methods proposed algorithms to increase the basis dimension, increasing the accuracy of the estimated correspondence.

Starting from a map with a reduced basis dimension k_{ini}, we can refine it, iteratively converting the correspondence into a larger map by increasing the basis dimension at each iteration with different strategies:

- **ZoomOut [Melzi et al. 2019b]:** iteratively computes maps and correspondences via Equations (2) and (4).
- **Iterative Meta Algorithm [Pai et al. 2021]:** converts between map via Equations (2) and (5). This method does not increase map accuracy since the conversion does not enforce additional constraints on the map.
- **Elastic [Hartwig et al. 2023]:** proposes a generalization of ZoomOut to elastic bases by iteratively alternating the conversions in Equations (2) and (7).

In this scenario, despite being based on the most theoretically sounded conversion, the Iterative Meta Algorithm shows limitations, since it is not regularized [Pai et al. 2021; Viganò and Melzi 2024]. ZoomOut and Elastic instead, are proven to minimize the following energy

$$E = \|C_{21} A_{12} - I\|_{HS} \tag{8}$$

where $\|\cdot\|_{HS}$ is the Hilbert-Schmidt norm [Hartwig et al. 2023], which in the case of orthonormal bases is equal to an energy on the orthonormality of C_{21}.

Considering LBO truncated basis, we can write Equation (4) in terms of the adjoint operator as

$$T_{12} = NS(\Phi_1^k, \Phi_2^k A_{12}). \tag{9}$$

As proven in [Melzi et al. 2019b], the solution of (9) minimizes:

$$E(\Pi_{21}, A_{12}) = \|\Pi_{21}\Phi_2^k A_{12} - \Phi_1^k\|, \tag{10}$$

where $\|\cdot\|$ indicates the matrix Frobenius-norm.

Looking at the bases as embeddings, the above discussions imply that we can consider A_{12} as a regularized way of aligning the embeddings minimizing Equation (10).

In the full basis, or for isometric shapes this map is the adjoint operator.

Moreover, we can extend the action of the adjoint operator from the operator that aligns the spectral embeddings to a map that aligns high-dimensional embeddings minimizing (10).

Our solution works in this direction, introducing a new functional representation that is capable of representing a broader space of correspondences relaxing the hypothesis of linearity of the map A_{12} between truncated embedding and optimizing it following regularized previous approach and extending them to new scenarios.

4 METHOD

In this paper, we focus for the first time on a neural representation of the operator that aligns the functional spaces defined on 2-dimensional surfaces embedded in $\mathbb{R}^3$. More specifically, we design a neural architecture to generalize the action of the adjoint linear operator [Huang and Ovsjanikov 2017], which has been proven to be the only functional operator that transfers delta distributions [Pai et al. 2021] and is thus preferable for shape matching. We organize the rest of this section as follows. We define our neural representation, analyze the role of non-linearity in the proposed solution, and show that it can adapt to different bases for the functional spaces. Finally, we propose a specific refinement technique designed to exploit the properties of our representation.

4.1 Neural Adjoint Maps

We propose *Neural Adjoint Map* (NAM), a novel representation to generalize, through a neural function, the action of the adjoint operator between functional spaces approximated by truncated embeddings.

We consider H defined as the following space of functions:

$$H = \{h : \mathbb{R}^k \to \mathbb{R}^k, \text{ such that} \tag{11}$$
$$h(y) = yA + \sigma\big(\sigma(yW^{(1)})W^{(2)}\big)W^{(3)}\},$$

where σ is an activation function, $A \in \mathbb{R}^{k\times k}$, $W^{(1)} \in \mathbb{R}^{k\times K}$, $W^{(2)} \in \mathbb{R}^{K\times K}$, and $W^{(3)} \in \mathbb{R}^{K\times k}$ are linear parametric transformations, with $K \neq k$.

Given two discrete shapes $\mathcal{X}_1$ and $\mathcal{X}_2$ represented by n_1 and n_2 points in $\mathbb{R}^3$ and their spectral embeddings $\Phi_1^k \in \mathbb{R}^{n_1\times k}$, $\Phi_2^k \in \mathbb{R}^{n_2\times k}$ truncated to dimension k, we can interpret H as a space of function between their spectral embeddings.

Leveraging the neural network solutions, we can parametrize the functions in H through an MLP. With this choice and assuming there is a correspondence Π_{21} between the points of the shapes, we can optimize for a neural function $h \in H$ that best aligns the embeddings Φ_1^k and Φ_2^k by minimizing the following energy

$$E_{NAM}(h) = \|\Pi_{21}h(\Phi_2^k) - \Phi_1^k\|. \tag{12}$$

We refer to the function $h \in H$ that minimizes this energy as the *Neural Adjoint Map* (**NAM**) associated with Π_{21}.

If we restrict the space of solutions in Equation (12) to the space of linear combination of Φ_2, this loss is minimized by $h(\Phi_2^k) = \Phi_2^k A_{12}$, where A_{12} is the adjoint operator of the functional map induced by Π_{21}, as we discussed in Equation (10). For this reason, if the spectral embeddings are the full bases or the two bases are linearly alignable (i.e. the two shapes are isometric), NAM corresponds to the action of the adjoint operator. In this sense, the NAM definition generalizes the notion of the adjoint operator. Instead, if we consider truncated bases and the shapes are not precisely isometric, the NAM gives rise to a broader space of solutions that will better align the embeddings, enabling more precise estimations of the correspondence, as we will see in the experimental Section.

As proven by the following proposition, under specific hypothesis, the minimization of Equation (10) induces the injectivity of the neural function h.

Proposition 4.1. *Given a function $g : \mathbb{R}^k \to \mathbb{R}^k$ such that $g(\Phi_1^k) = \Pi_{21}\Phi_2^k$, then a NAM h realizes $E_{NAM}(h) = 0$ in Equation (12) if and only if*

$$h \circ g = I_1 \tag{13}$$

where I_1 is the identity function on Φ_1^k.

We report the proof in the supplementary materials. Considering the full LBO basis or isometric shapes, this proposition generalizes to NAM the analogy between Equation (8) and Equation (10) described in Section 3.2.

Point to Point conversion. Given a permutation Π_{21}, we get a corresponding NAM by estimating the MLP that simply minimize E_{NAM} from Equation (12) as loss. In the opposite direction, given a NAM h, we can recover the correspondence by performing the nearest search (NS) in the space of the spectral embedding of $\mathcal{X}_1$:

$$T_{12} = NS(\Phi_1^k, h(\Phi_2^k)). \tag{14}$$

This conversion method resembles its linear counterpart defined in Equation (9).

4.2 The role of non-linearity

Crucial in our representation is the non-linear component that defines the neural map (11). Indeed, assuming the linearity of the adjoint map between truncated embedding is equivalent to imposing some prior on the map that it represents. In the case of spectral embedding, this prior is the isometry. Indeed it holds that

Proposition 4.2. *Let's assume to have a k dimensional embedding $\Xi_1 \in \mathbb{R}^{n\times k}$. There will always be a $\Xi_2 \in \mathbb{R}^{n\times k}$, in a (1:1) correspondence with Ξ_1, such that Ξ_1 and Ξ_2 are not linearly alignable i.e $\nexists A \in \mathbb{R}^{k\times k}$ s.t. $\Xi_1 = \Xi_2 A$.*

This is trivial, considering that the space of linear transformation of Ξ_1 is equivalent to $\mathbb{R}^{k\times k}$ which is strictly contained in $\mathbb{R}^{n\times k}$ which is the space of possible embedding of the given dimensions. If we consider spectral embeddings, it holds that

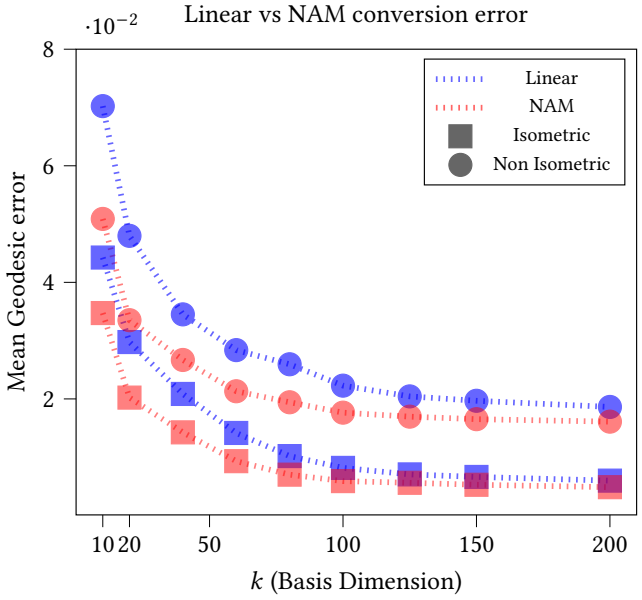

Fig. 2. Comparison between linear and NAM representation of ground truth correspondence. On the x-axis the basis dimension k. On the y-axis the Mean geodesic error of the recovered map. The introduction of the non-linear module induces a better alignment of the basis and consequently a better correspondence estimation. This result is more evident for non-isometric pairs. We note that as we increase the basis dimension the difference between the NAM representation and the linear one decreases.

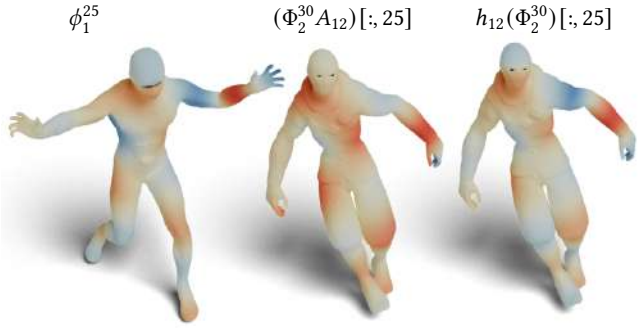

Fig. 3. 25th basis function of X_1 (left) transferred with different approaches (linear in the middle and NAM on the right). We can see that NAM better recovers the basis function on the target enabling a more precise alignment.

Proposition 4.3. *Φ_1^k and Φ_2^k are linearly alignable for any $k < n$ iff the shapes are isometric.*

We prove this Proposition in the supplementary material. This Proposition implies that if we convert a correspondence into a linear functional operator that acts between two spectral embeddings, we can aim to recover the correspondence only if we are implicitly assuming the correspondence to be an isometry. Considering a NAM, we can overcome this limitation thanks to its non-linear component. Indeed, if we consider two truncated spectral embeddings of dimension k for $\mathcal{X}_1$ and $\mathcal{X}_2$, assuming that they are linearly alignable through the action of an adjoint operator is equivalent to assume that:

$$\{\phi_1^i\}_{\{i=0,..,k\}} \subset span(\{\Pi_{21}\phi_2^i\}_{\{i=0,..,k\}}). \tag{15}$$

This means that the basis functions of Φ_1^k can be described by a linear combination of $\Phi_{21}\Phi_2^k$. However, the frequencies spanned by different truncated spectral embedding can vary highly, even more in the case of non-isometry. For this reason, looking for the space of linear combinations of the embedding dimensions of $\Pi_{21}\Phi_2^k$ cannot give rise to a precise description of Φ_1^k. However, if we consider $H(\Phi_2^k) = \{h(\Phi_2^k), \quad h \in H\}$, we have

$$span(\Phi_2^k) \subset H(\Phi_2^k) \tag{16}$$

and, since both NAM and linear transformation acts row-wise

$$span(\Pi_{21}\Phi_2^k) \subset H(\Pi_{21}\Phi_2^k) = \Pi_{21}H(\Phi_2^k) \tag{17}$$

where $\Pi_{21}H(\Phi_2^k) = \{\Pi_{21}h(\Phi_2^k), \quad h \in H\}$.

In other words, the space generated by $H(\Phi_2^k)$ is broader than the span of Φ_2^k. For this reason, NAM yields to better capture the image of deltas described by the spectral embeddings of the source shape, and thus, any correspondence is better represented by NAM than by a linear map.

To show some examples of the advantages provided by NAM, we perform some experiments on the Deforming Things [Magnet et al. 2022] dataset (DT4D), which is split into isometric and non-isometric pairs (we refer to Section 5 for more details on the dataset). In Figure 2, we show that NAM is better than the linear alternative for representing correspondences. For different values of k (basis dimension on the x-axis), we encode a ground truth correspondence with the adjoint linear representation (blue) and with our NAM (red). Then we convert them back to a point-to-point map respectively with Equations (9) and (14). On the y-axis, we report the mean geodesic error after the conversion computed as the geodesic distance between the estimated match and the ground truth one on average on all the points and all the isometric (square) and non-isometric (circle) pairs. We can see that with NAM, we consistently obtain more accurate correspondences for all k. Furthermore, as expected, the gap is more evident in non-isometric pairs.

In Figure 3, we report a comparison in the transfer of a function of the LBO basis of $\mathcal{X}_1$, more specifically ϕ_1^{25} (on the right), to $\mathcal{X}_2$ using the linear approach (middle) or NAM (right) both exploiting bases of size 30 Φ_1^{30} and Φ_2^{30}. The colormap encodes the values of the function: white is zero, blue is negative and red is positive. $\mathcal{X}_1$ and $\mathcal{X}_2$ are non-isometric, and we can see that the linear approach loses many details while NAM is recovering the function more precisely. This ability to recover the transferred signal coincides with better alignment of the embeddings from which we estimate the correspondence.

4.3 NAM Across Embeddings

We defined and motivated NAM by considering the LBO basis as embedding. While this embedding is the most frequent choice in the context of shape matching due to its smoothness and hierarchical structure, it often struggles to represent high-frequency signals and does not perform well if the shapes are strongly non-isometric.

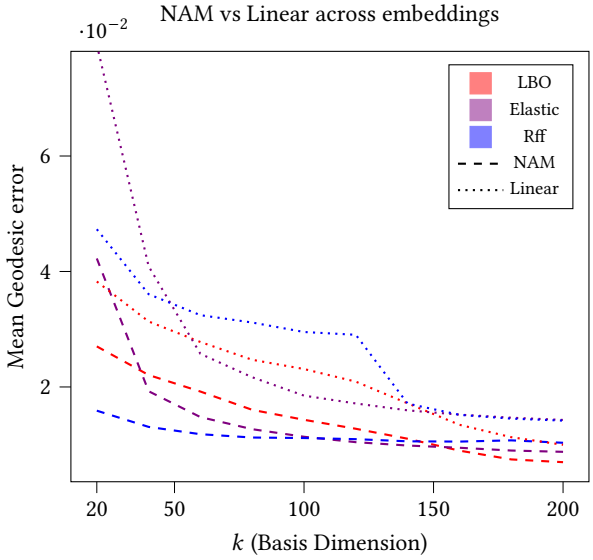

Fig. 4. Our approach applied to different embeddings with different dimensionalities (x-axis). The NAM representation (dashed lines) returns better maps than the linear representation (dotted lines) for any embedding at any basis dimension.

These limitations gave rise to the exploration of alternative embeddings in many applications.

The definition of NAM can be applied universally across various high-dimensional embeddings. In particular, among many possibilities, we consider the following embeddings:

- Elastic Basis: Introduced in [Hartwig et al. 2023], these bases are computed as the eigenfunctions of an elastic operator that models elastic properties of shapes, such as bending and stretching. They have been proven to represent high-frequency signals and can be used in Functional maps-like pipelines. However, this basis often struggles to represent the global property of the shapes, and they are numerically unstable.
- Random Fourier Features (RFF): Introduced in [Tancik et al. 2020], RFF are samplings of Fourier Radial Basis functions. These embeddings are extrinsic and have been proven to be well-suited for the efficient computation of high-frequency neural functions. However, these embeddings can be sensitive to topological issues because their computation is coordinate-based.

Figure 4 illustrates a comparison between the linear (dotted lines) and NAM (dashed lines) representations when applied to different embeddings with different dimensions (k on the x-axis), including LBO basis (red), elastic basis (purple), and RFF (blue). The NAM representation consistently produces better maps than linear representation.

Finally, NAMs also work effectively with other embeddings, such as learned embeddings or direct 3D coordinates. In the Results section, we demonstrate the impact of NAM in such cases. Furthermore, our definition aligns well with the concept of Neural Operator, recently introduced in [Kovachki et al. 2024].

We remark that we build the notion of NAM inspired by the action of the adjoint operator in the case of spectral embeddings. [Hartwig et al. 2023] extends the ZoomOut algorithm to any basis. Similarly, we could build NAM considering slightly different formulations. In supplementary materials, we discuss this issue and propose a variant of NAM that generalizes to any basis. However, this definition requires selecting an embedding with linear independent dimensions and thus does not generalize to all embeddings, as our current definition does.

4.4 Neural ZoomOut

Considering the NAM representation, we design a refinement algorithm to improve the accuracy of an estimated correspondence. We design an iterative algorithm, namely Neural ZoomOut (NZO), exploiting the NAM representation and following previous methods described in Section 3.2. NZO iteratively converts a correspondence into an NAM and then recovers the correspondence via the nearest search, augmenting the embedding dimension in the conversion to an NAM at each iteration as described in Algorithm 1.

ALGORITHM 1: Neural ZoomOut (NZO)

1: Input: Π_{21}
2: **for** $k_{ini} \leq k \leq k_{end}$ **do**
3: Optimize for h minimizing Equation (12)
4: Compute $\Pi_{21} = NS(\Phi_1^k, h(\Phi_2^k))$
5: **end for**
6: Output: Π_{21}

By construction Neural ZoomOut, is a generalization of ZoomOut [Melzi et al. 2019b] that extends to any embedding. In particular, is trivial to show that if we restrict the space of solution of (12) to the space of linear mapping $\Phi_2^k A$, then $(\Phi_1^k)^\dagger \Pi_{21} \Phi_2^k A = (\Phi_1^k)^\dagger \Phi_1^k$ which is $C_{21}A = (\Phi_1^k)^\dagger \Phi_1^k = I$. This means that in the linear case, minimizing for h is equal to computing the right inverse of C_{21}. If we have two isometric shapes with orthonormal bases, this is equal to computing the adjoint $C_{21}^\top$, so NZO coincides with ZoomOut in the case of isometry. Combining the ZoomOut procedure with the representation capabilities of the non-linear module of a NAM, we get a flexible alternative to the Zoomout algorithm that we can apply to other embeddings.

Alternating Neural ZoomOut. The flexibility of NZO is intriguing, but with the possible unstructured nature of other embeddings, defining a notion of hierarchy to guide the iterative process of NZO could be problematic. For this reason, we define a new algorithm that can alternately change the selected embedding to leverage different representations. At each process iteration, we convert and recover the point-to-point map in a NAM twice: the first time with the spectral embedding Φ and the second with another embedding F as outlined in Algorithm 2.

Despite its simplicity, this new refinement, namely Alternating Neural ZoomOut (NZO*), refines any correspondence, merging different properties from different embeddings. If the second embedding adopted is extrinsic, it can substitute for an additional explicit

Table 2. Comparison of refinement methods NZO and NZO* on different embeddings. On the diagonal, we highlight the entries that refer to NZO. For NZO*, on the rows, we have the first embedding selected (lines 3 and 4 of Algorithm 2) and on the columns the second one (lines 5 and 6 of Algorithm 2). For each refinement, we report the ratio between the average geodesic error of the final refined map (Final) and the one of the initialization and the mean value of this ratio in all the iterations (Mean) (smaller is better).

	LBO		Elastic		RFF	
	Final	Mean	Final	Mean	Final	Mean
LBO	**0.72**	0.78	0.83	0.87	0.752	**0.77**
Elastic	1.23	1.37	1.49	1.70	1.12	1.19
RFF	0.83	0.87	0.90	0.93	0.90	0.93

ALGORITHM 2: Alternating Neural ZoomOut (NZO*)

1: Input: Π_{21}
2: **for** $k_{ini} \leq k \leq k_{end}$ **do**
3: Optimize for a NAM h minimizing Equation (12)
4: Compute $\Pi_{21} = NS(h(\Phi_2^k), \Phi_1^k,)$
5: Optimize for a NAM $\tilde{h}$ minimizing Equation (12) with another embeddings F_1, F_2
6: Compute $\Pi_{21} = NS(\tilde{h}(F_2), F_1)$
7: **end for**

extrinsic regularization. However, since our representation is general to any embedding, it can be applied to 3D coordinates and optimized with additional energies, such as Chamfer distance, elastic energies, and Dirichlet smoothness losses, adding them as losses in the optimization of the NAM. We leave these explorations as future directions.

In Table 2, we report the performance of NZO and NZO* for 50 non-isometric shape pairs from DT4D, exploiting different embeddings. We obtain all the results from an initial correspondence extracted from a ground truth functional map of dimension 20 encoded in the spectral embeddings. NZO on the spectral embedding (LBO) achieves the best result in the final map, while the combination of LBO and RFF with NZO* has the best results in mean on the iterations. NZO and NZO* will refer to these two configurations when not explicitly said. From this Table, we can infer that the spectral embedding is the best choice in NZO* for lines 3 and 4 in Algorithm 2.

5 EXPERIMENTS

In this section, we demonstrate the versatility and effectiveness of our method for refining correspondences in a broad range of scenarios. Furthermore, we assess the robustness of the proposed solution to different initializations, showing that it is the only technique to reduce the error in any case from very accurate to noisy initializations. Finally, we ablate on our design choices, showing the role of each component of our solutions.

5.1 Experimental setting

Datasets. To prove the flexibility of our method, we select different datasets that cover some difficult scenarios in which classical functional solutions often fail. In particular, we consider near-isometric meshes, non-isometric meshes, and point clouds. We consider the following datasets:

- **SHREC19**: a dataset of near-isometric human meshes with different connectivity [Melzi et al. 2019a].
- **DT4D**: a dataset of triangular meshes representing humanoid subjects, divided into intra-class pairs with isometric deformation (**DT4D intra**) and non-isometric inter-class pairs (**DT4D inter**) pairs [Magnet et al. 2022].
- **SMAL**: composed of meshes that represent quadrupeds from different classes in varying poses with large non-isometric deformations [Zuffi et al. 2017].
- **FAUST scan**: a collection of point clouds obtained from human scans [Bogo et al. 2014].
- **KINECT**: a new dataset of 15 point clouds extracted from the BEHAVE dataset [Bhatnagar et al. 2022]. This dataset is composed of Kinect acquisitions of humans interacting with objects.

In the case of meshes, we evaluate our performance on subsets of the datasets, considering the test set coming from the train-test split which arises from previous experiments in the literature, and more specifically, we consider the one from [Cao et al. 2023].

For the FAUST scan dataset, composed of point clouds, we selected the last 20 shapes of the training set [Bogo et al. 2014], for which is available the ground truth registration with the SMPL template [Loper et al. 2015]. Finally, the KINECT dataset is composed of 15 point clouds, for which an SMPL Mocap registration is provided. This lets us recover a ground truth to evaluate the maps. Details on how we built this test set on the Supplementary material. We note that we consider point clouds composed of $10K$ points randomly sampled from the original data.

Evaluation metrics. In our experiments, we evaluate the effectiveness of our methods by considering an accuracy metric. For every shape couple, we compute the accuracy of the estimated point-to-point correspondence T_{12} with respect to the given ground truth T_{12}^{gt} using the standard evaluation method from [Kim et al. 2011]. Since we compare both intrinsic and extrinsic methods, we report the error in terms of the geodesic distance (Geod) and the Euclidean distance (Eucl).

Moreover, to evaluate the improvement achieved by a refinement technique with respect to the initialization, we compute the improvement ratio r, defined as the difference between the error of the input and output maps of the refinement methods.

$$r = (Geod_{in} - Geod_{out})/Geod_{in} * 100\,, \tag{18}$$

This metric is a relative measure of the performance of each competitor.

Implementation details. The official implementation can be found at https://github.com/gviga/NAM-NeuralAdjointMaps. Our method is implemented in PyTorch [Paszke et al. 2019] and, given its simplicity, does not depend on any particular component or additional library. The optimization of NAM is performed via gradient descent with ADAM optimizer [Kingma and Ba 2017]. In the upsampling phase, we consider $k_{ini} = 20$ and $k_{end} = 200$, with a step of 20. The nonlinear module is composed of 2 layers with a width of $K = 128$.

Table 3. **Shape Matching**: Comparison of different refinement methods across diverse scenarios. The best result is highlighted in bold, and the second best is in Blue. Isometric results (left), Non-Isometric results (middle), and Pointclouds results (right). In the last column, we report the mean improvement of the geodesic error between the datasets considered.

	Near-isometric						**Non-isometric**						**Pointclouds**					
	SHREC19			DT4D intra			SMAL			DT4D inter			FAUST scan			KINECT		
	Geod	Eucl	r	Geod	Eucl	r	Geod	Eucl	r	Geod	Eucl	r	Geod	Eucl	r	Geod	Eucl	r
INI	7.88	7.37	0	0.90	0.75	-	7.10	5.45	-	4.52	1.42	-	14.18	10.1	-	34.2	34.7	-
ZO	6.51	5.01	17	1.06	0.71	-18	6.02	4.54	15	5.33	2.35	-18	13.3	9.4	6	56.4	95.9	-65
NDP	9.88	5.78	-25	3.87	3.01	-319	7.74	2.9	-9	6.67	1.74	-48	13.7	**8.2**	3	30.4	**22.9**	13
ULRSM ref	**4.66**	**2.8**	**40**	**0.85**	**0.5**	**6**	**3.87**	**0.99**	**45**	6.68	5.24	-48	NA	NA	NA	NA	NA	NA
NZO	6.66	5.23	15	0.99	0.64	-10	4.39	1.23	38	**4.12**	**1.05**	**9**	12.9	8.7	9	29.3	26.3	14
NZO*	6.64	5.13	16	1.97	2.6	-119	3.97	1.01	44	4.20	1.16	7	**12.8**	8.6	**10**	**28.3**	25.2	**17**

We perform an ablation on these choices in the supplementary. The LBO eigenfunctions are computed using the cotangent weights scheme [Pinkall and Polthier 1993] for meshes and robust laplacian [Sharp and Crane 2020] for point clouds. For initializations and competitors, we consider official implementations and model checkpoints provided by the authors.

5.2 Shape matching

In Table 3 we gather our quantitative evaluation on all the datasets, considering the scenarios of near-isometric meshes, non-isometric meshes, and point clouds.

In the context of near-isometric and non-isometric shape matching, the state-of-the-art for computing correspondences are represented by learned approaches. The works of [Cao et al. 2023; Sun et al. 2023] represent a de-facto standard for shape matching. Both methods are based on learning universal features regularized by functional maps modules. In the case of triangular meshes, we consider [Cao et al. 2023] to initialize the maps. In the case of Pointcloud, we consider the method proposed in [Cao and Bernard 2023], which learns universal features for point clouds considering mesh information at training time.

We compare our methods with three representative methods: a fully intrinsic method, the Zoomout algorithm (ZO)[Melzi et al. 2019b], a fully extrinsic method, the Deformation Pyramid (NDP) algorithm [Li 2022], and, for meshes, the refinement method specifically designed for the initialization [Cao et al. 2023], that we indicate with (ULRSM ref). We note that other functional-map based refinements [Magnet et al. 2022; Pai et al. 2021; Ren et al. 2021], which derive from [Melzi et al. 2019b], could be redefined following our new definition.

Near-isometric meshes. In the case of near-isometric meshes, the performance of NZO is in line with the ZO algorithm. These results validate our previous analysis since the effect of NAM is negligible in the case of isometric pairs, and it could slightly decrease the performances given the harder optimization induced by the larger number of parameters. Thus, the linear representation and the energy minimized by our algorithm are those of ZO. Instead, ULRSM ref returns the best accuracy, minimizing additional smoothness constraints. We note that in the DT4D dataset, NZO decreases the accuracy of the map. We stress that this unique behavior is given by the extreme initialization quality, which is so accurate that its projection to low frequency can only decrease its quality. In this scenario, we should choose a bigger initial basis dimension. We show the effect of this choice in the supplementary material.

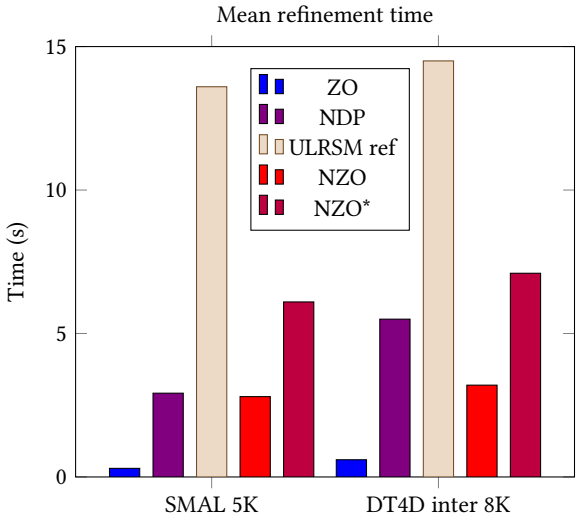

Fig. 5. Mean refinement time for each method on two datasets with different numbers of points, indicated in the x-axis.

Non isometric meshes. In more challenging scenarios the results indicate the need to consider our refinement method. The performances of NZO and NZO* are comparable to the ones of ULRSM ref in the SMAL dataset. However, our methods are the only ones that can improve the correspondence estimated on the DT4D inter dataset. Indeed, the alternatives even decrease the accuracy of the initial maps in this scenario, showing less flexibility than our solutions. Moreover, in Figure 5, we consider the mean time spent refining a map. We notice that NZO is the second fastest method. As expected it is slower than ZO, given the more parameters that need to be optimized. Our method is comparable to NDP, however, it seems that it scales better with the number of points of the shapes. However, most notably, it takes 75% less time to perform refinement than ULRSM ref [Cao et al. 2023].

Point Cloud. If we consider unstructured data, the challenge of computing correspondences is even more challenging because of the lack of geometrical information. Having a good refinement is crucial if we consider that many trained approaches do not scale

Table 4. Refinement method applied to different initialization. ULRSM refers to the method of [Cao et al. 2023]. One shot refers to the same method in which the basis dimension is 30 and the training has been only conducted for 1 epoch. Diff3D refers to [Dutt et al. 2024]. GT20x20 refers to a map computed from a ground-truth functional map of dimension 20.

	GT(20x20)	ULRSM	ULRSM One-shot	Diff3D
Init	4.27	7.10	9.21	31.8
ZO	8.29	6.02	6.45	24.7
NDP	5.17	7.74	8.58	25.1
ULRSM ref	NA	**3.87**	**4.98**	NA
NZO	3.5	4.39	5.18	22.6
NZO*	**3.0**	3.97	5.12	**22.0**

Table 5. Refinement method applied to different initialization. SSMSM refers to the method of [Cao and Bernard 2023]. Lepard refers to the supervised method of [Li and Harada 2022]. NearestNeighbour refers to a map computed performing nearest neighbor between the 3d coordinates of the shapes. GT20x20 refers to a map computed from a ground-truth functional map of dimension 20.

	GT(20x20)	SSMSM	Lepard	NearestNeighbour
Init	5.43	14.18	17.5	31.5
ZO	41.5	13.3	17.8	30.3
NDP	7.01	13.7	13.9	29.3
NZO	3.1	12.9	13.8	29.6
NZO*	**2.97**	**12.8**	**12.9**	**29.1**

Table 6. Performance of our method with different embeddings. For each representation, we consider a refinement specifically designed for it. For Elastic, the baseline is their iterative refinement, for LIE, the baseline is [Viganò and Melzi 2024]. NIE does not present a refinement in its original work, and in this case we consider $k_{ini} = 10, k_{end} = 20$.

	Dataset	Init	Baseline	NZO
ELASTIC [Hartwig et al. 2023]	ELASTIC	14.34	2.37	**1.69**
NIE [Jiang et al. 2023]	FAUST	14.24	NA	**12.8**
LIE[Marin et al. 2020]	FAUST 1k	2.33	1.95	**1.84**

well to high-resolution point clouds and are often optimized for point clouds with a fixed number of points, such as [Lang et al. 2021; Li and Harada 2022].

From our experiments, we note that our approach returns the best geodesic error in both the datasets considered. Moreover, we note how using extrinsic features highly improves the results of our method. This outcome highlights that having a method that works both with intrinsic and extrinsic representation is the best alternative in the case of point clouds, where the intrinsic operator's computation can be unstable and noisy. Indeed, we stress that our method, as well as other spectral approaches, requires a reasonable estimate of the Laplace-Beltrami eigenfunctions, or other surface-aware representations, which in turn depend on a good local surface approximation. However, it can also return good estimates in these challenging scenarios compared to the previous approaches.

However, we need to note that, differently from the mesh scenario, the initializations considered do not return accurate results. Our method can refine the correspondences considered but the problem of having robust initializations for pairs of point clouds is still an open problem.

5.3 Robustness

Different initializations . Other than [Cao et al. 2023], our method can be applied as a refinement of other methods to find correspondences. To test this, we consider alternative possible initializations to assess the robustness of our method in refining *any* correspondence. In Table 4 and in Table 5, we consider respectively the SMAL datasets and the FAUST Scan dataset with different possible initial correspondences. Our method improves any initial correspondence, showing unprecedented flexibility. In particular, the result of refining a ground-truth map $20x20$ indicates that, in the presence of a good refinement, the problem of shape matching reduces to the problem of aligning very compact representations in challenging scenarios.

Different embeddings . We assess the robustness of our method by considering other embeddings. Recently, different works proposed to substitute the LBO basis with other bases or learned embeddings; some methods propose a compatible refinement stage. For instance, learned embeddings such as LIE [Marin et al. 2020] and NIE [Jiang et al. 2023] have been proposed as a valuable alternative to the Laplacian basis. For the first, [Viganò and Melzi 2024] proposed a dedicated refinement. Moreover, we consider the elastic basis proposed by [Hartwig et al. 2023] and its refinement. Our method can be extended to these bases, showing an unprecedented generalization capability.

We replicate some of the experiments proposed in these works. In Table 6, we show our results. Adapting our refinement to the proposed embeddings, we can compare its performance with algorithms (indicated as baseline) designed explicitly for the considered representation. Also in this case we can obtain good refinement, showing that the NAM representation and NZO refinement sets themselves as a universal framework that can be applied to any high-dimensional embedding.

5.4 Ablation

We conducted ablations on 50 random pairs from the SMAL dataset To show the role of each choice that we made to implement the NAM representation and the NZO algorithm.

The goal of NAM is to provide a smooth invertible representation of maps between shapes. We compare our approach to other possible alternatives, which include: (A) learning a neural deformation field between the 3D points of the shapes and (B) learning two forward and backward deformation fields explicitly imposing a bijectivity loss. In the first plot of Figure 6, we show the geodesic error of recovered maps during the optimization of neural functions to represent a ground-truth correspondence. We note that working between functional representations results in more efficient and regular optimizations. On the other hand, working with 3d embeddings, we can impose additional regularizations, like volume preservation or non-rigidity terms. However, this would lead to a more expensive optimization, which is not ideal for refinement.

We further evaluate key design choices of NAM, such as the impact of the non-linear module and its activation function. Figure 6 shows that considering a hybrid model is the only way to obtain

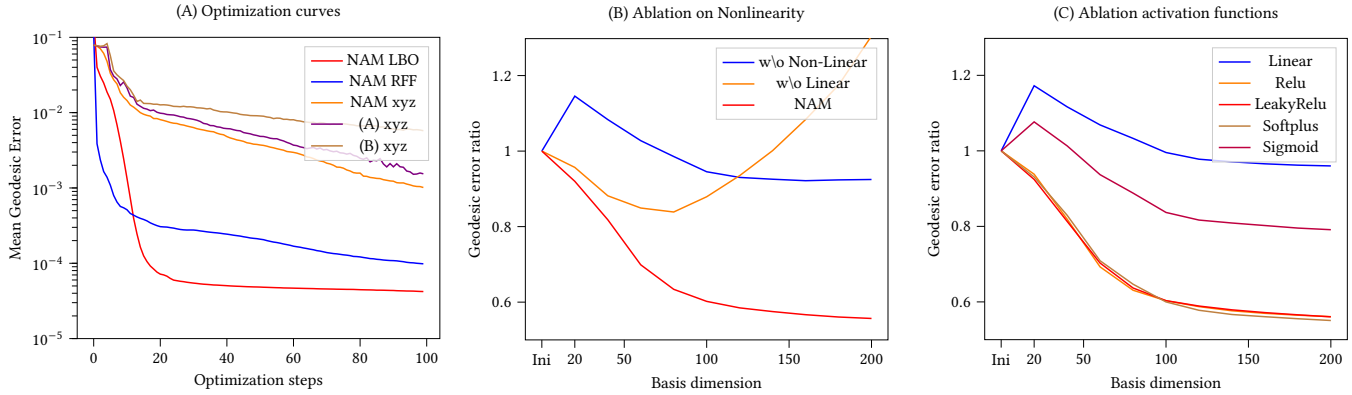

Fig. 6. Ablation studies on NAM. (Left) Conversion errors at different stages of the optimization. (Center) Linear vs. non-linear modules, showing both modules are necessary for effective refinement. (Right) Ablation on the nonlinear function: ReLU-like activations yield the best results.

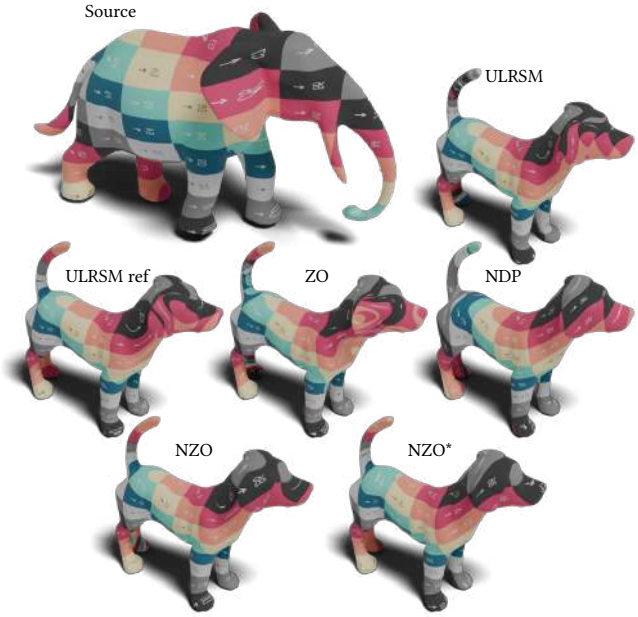

Fig. 7. Texture transfer on a strong non-isometric pair from the SHREC20 dataset [Dyke et al. 2020]. We note that ZO algorithm fails to transfer the signal between regions in which the geometry changes a lot, e.g., the face. At the same time, NDP fails in regions in which there is a strong pose variation, e.g., the tail. Our method instead returns an overall smooth transfer.

an accurate refinement. ReLU-like activations outperform other non-linearities, as shown in Figure 6. In the supplementary, we report additional details on the ablations and on the hyperparameter choices of the model.

6 APPLICATIONS

6.1 Signal transfer

A typical application of shape matching is the transfer of signals between two shapes. Indeed, let us assume that we have a signal on the surface of a mesh or a point cloud. The signal can be defined as a function $\psi_2 : \mathcal{L}^2(X_2)_2 \to \mathbb{R}^d$, where d is the dimension of the signal, e.g. 3 for deformations, 2 for UV maps, 3 for colors or 1 for scalar functions. If we have access to the permutation Π_{21}, the transferred signal can be defined as $\psi_1 = \Pi_{21}\psi_2$. Using a functional map can be useful to obtain a smooth signal transfer. Indeed, we can directly map the signals' coefficient using a C_{21} as

$$\psi_1 = \Phi_1^k C_{21} (\Phi_2^k)^\dagger \psi_2. \tag{19}$$

This is convenient, especially when the shape discretization varies between the shapes, and we want a smooth signal transfer. This functional map can be computed between any embedding, for instance, with NZO we report transfer with functional maps on LBO eigenfunctions. For NZO* solution, instead, we transfer the signal considering the RFF as embeddings in (19).

Texture transfer. In Figure 7 and 8, we show the performances of the refinements in transferring textures between strong non-isometric pairs from the SHREC20 [Dyke et al. 2020], DT4D [Magnet et al. 2022] and SMAL [Zuffi et al. 2017] datasets. Our methods return texture transfers that present improved accuracy in high-frequency parts of the shapes. Interestingly, focusing on the first example proposed, we can see that the map refined by NZO presents a high-frequency artifact that comes from an error in the initialization (in the forehead of the target shape). This highlight the fact that a NAM can represent high-frequency signals.

Coordinate transfer and registration. As textures, we can transfer coordinates to perform registration. In these scenarios, linear models often struggle since the spectrum does not represent high-frequency extrinsic signals. For this reason, with NAM, we can leverage Randomized Fourier Features. In Figure 9 we show a qualitative example of our coordinate transfer. We note that NZO* outperforms NZO in

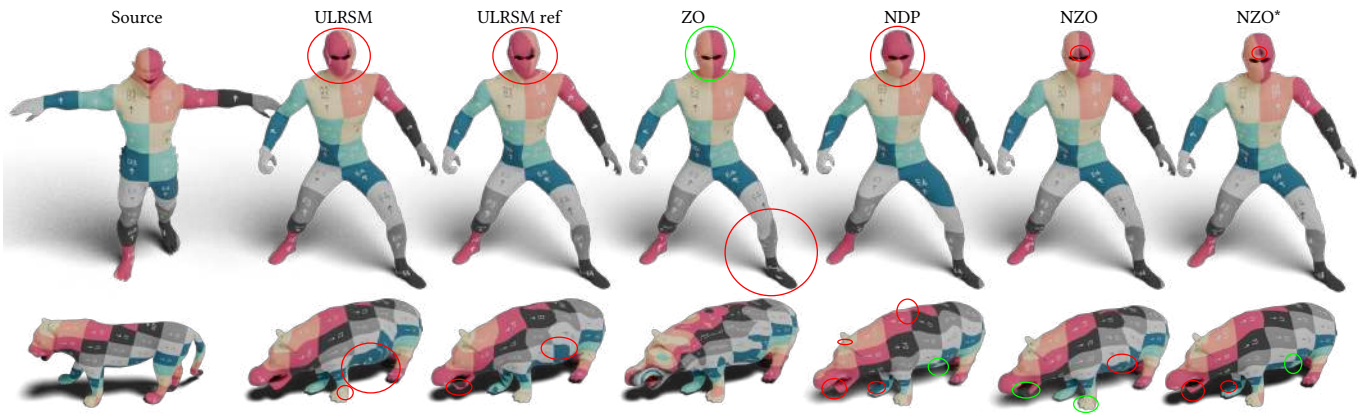

Fig. 8. Texture transfer for different refinement methods. The red circles indicate notable errors, green circles indicate improved mapping.

terms of smoothness of the transfer and precision in the protrusion of the point clouds. We stress that our method does not explicitly minimize registration losses like chamfer distance. However, it can still recover a lot of information and get results comparable with methods that explicitly does, such as NDP.

6.2 Neural Field Transfer

In recent works, neural field-based parametrization of signals has become a standard in the community. Having a smooth neural representation of the correspondence is particularly useful in this scenario. Moreover, our representation is convenient not only for its neural nature but also since it is compatible with Randomized Fourier features (RFF) and the Laplace Beltrami operator. This design choice makes it ideal for transferring neural fields. RFF and LBO eigenfunctions are indeed two of the most common domains in which neural fields are defined [Koestler et al. 2022; Tancik et al. 2020].

Deformation Transfer. We can consider deformation transfer as an example of this scenario. Let us assume we receive as input the shape A with a neural deformation field d_A defined on its RFF or LBO eigenfunctions. If we assume to have an additional deformed shape B, we can transfer the deformation field from A to B by simply combining a NAM h with the neural field $d_B = d_A \circ h$. In Figure 10, we show this example on the FAUST Scan dataset [Bogo et al. 2014]. We note that intrinsic methods fail to accurately transfer the deformation between the protrusions of the shapes, while NZO*, mapping extrinsic representation can obtain near-optimal results.

7 Conclusions

In this work, we designed a new neural representation for shape correspondences that extends the properties of the classical functional maps to work with any embedding, even in challenging scenarios.

Limitations and Future Works. However, the quality of the recovered correspondences still depends on the properties of the embeddings. For example, the performances of our NZO* still suffer from topological noise. Moreover, despite being quite efficient compared with neural alternatives, its performance can be improved by considering other optimization strategies, such as the one implemented in [Li 2022]. Moreover, we implement the new NAM representation only in the context of refinement. An intriguing future direction would be to integrate NAM in learning pipelines during the training phase to learn better features, as in [Donati et al. 2020] or better embeddings, as in [Marin et al. 2020].

In conclusion, NAM is a neural solution that extends recently proposed alternatives for functional representation and refinement of shape correspondences. Evaluations on standard benchmarks and challenging datasets demonstrate that this approach achieves state-of-the-art accuracy. NAM can handle isometric cases and generalize to model other deformations among shapes. They capture intricate geometric details that conventional linear approaches often overlook. NAM can be applied to various embeddings representing 3D shapes, can refine any input correspondence, and is effective across different scenarios, including isometric meshes, non-isometric meshes, and point clouds.

Acknowledgments

The MUR supported this work with the grant “Dipartimenti di Eccellenza 2023-2027” REGAINS of the Department of Informatics, Systems and Communication of the University of Milano-Bicocca and by the PRIN project GEOPRIDE Prot. 2022-NAZ-0115, CUP H53D23003400001. This work was partially supported by NVIDIA through the Academic Hardware Grant for the project “Learned representations for implicit binary operations on real-world 2D-3D data”, by the ERC Consolidator Grant 101087347 (VEGA), the ANR AI Chair AIGRETTE, and by gifts from Ansys and Adobe Research.

References

Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes. arXiv:2205.02904 [cs.GR] https://arxiv.org/abs/2205.02904

Souhaib Attaiki and Maks Ovsjanikov. 2023. NCP: Neural Correspondence Prior for Effective Unsupervised Shape Matching. arXiv:2301.05839 [cs.CV] https://arxiv.org/abs/2301.05839

Omri Azencot, Anastasia Dubrovina, and Leonidas Guibas. 2019. Consistent Shape Matching via Coupled Optimization. *Computer Graphics Forum* 38, 5 (2019), 13–25.

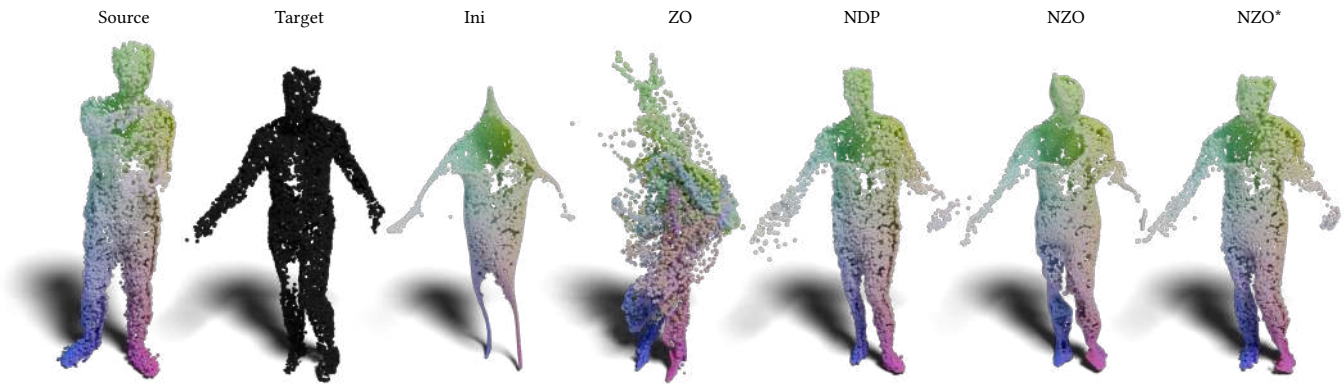

Fig. 9. Coordinate transfer on KINECT. We consider results obtained in the refinement of a ground-truth functional map 20x20 indicated as Ini.

Bharat Lal Bhatnagar, Xianghui Xie, Ilya A. Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. 2022. BEHAVE: Dataset and Method for Tracking Human Object Interactions. arXiv:2204.06950 [cs.CV] https://arxiv.org/abs/2204.06950

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. 2014. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*. IEEE, Columbus, Ohio, 3794–3801.

Dongliang Cao and Florian Bernard. 2023. Self-Supervised Learning for Multimodal Non-Rigid 3D Shape Matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dongliang Cao, Paul Roetzer, and Florian Bernard. 2023. Unsupervised Learning of Robust Spectral Shape Matching. *ACM Trans. Graph.* 42, 4, Article 132 (jul 2023), 15 pages. doi:10.1145/3592107

Enric Corona, Gerard Pons-Moll, Guillem Alenyà, and Francesc Moreno-Noguer. 2022. Learned Vertex Descent: A New Direction for 3D Human Model Fitting. arXiv:2205.06254 [cs.CV] https://arxiv.org/abs/2205.06254

Bailin Deng, Yuxin Yao, Roberto M. Dyke, and Juyong Zhang. 2022. A Survey of Non-Rigid 3D Registration. arXiv:2203.07858 [cs.CV]

Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. 2023. SE-ORNet: Self-Ensembling Orientation-aware Network for Unsupervised Point Cloud Shape Correspondence. arXiv:2304.05395 [cs.CV] https://arxiv.org/abs/2304.05395

Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. 2022. Deep Orientation-Aware Functional Maps: Tackling Symmetry Issues in Shape Matching. arXiv:2204.13453 [cs.CV]

Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. 2020. Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8592–8601.

Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J. Mitra. 2024. Diffusion 3D Features (Diff3F): Decorating Untextured Shapes with Distilled Semantic Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4494–4504.

Roberto M. Dyke, Yu-Kun Lai, Paul L. Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. 2020. SHREC'20: Shape correspondence with non-isometric deformations. *Computers & Graphics* 92 (2020), 28–43. doi:10.1016/j.cag.2020.08.008

Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. 2020. Deep Shells: Unsupervised Shape Correspondence with Optimal Transport. arXiv:2010.15261 [cs.CV]

Dvir Ginzburg and Dan Raviv. 2021. Dual Geometric Graph Network (DG2N) Iterative Network for Deformable Shape Alignment. In *2021 International Conference on 3D Vision (3DV)*. IEEE. doi:10.1109/3dv53792.2021.00141

Florine Hartwig, Josua Sassen, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. 2023. An Elastic Basis for Spectral Shape Correspondence. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) *(SIGGRAPH '23)*. Association for Computing Machinery, New York, NY, USA, Article 58, 11 pages. doi:10.1145/3588432.3591518

Ruqi Huang and Maks Ovsjanikov. 2017. Adjoint Map Representation for Shape Analysis and Matching. *Computer Graphics Forum* 36, 5 (2017), 151–163.

Puhua Jiang, Mingze Sun, and Ruqi Huang. 2023. Neural Intrinsic Embedding for Non-rigid Point Cloud Matching. arXiv:2303.01038 [cs.CV] https://arxiv.org/abs/2303.01038

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. 2011. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 79.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] https://arxiv.org/abs/1412.6980

Lukas Koestler, Daniel Grittner, Michael Moeller, Daniel Cremers, and Zorah Lähner. 2022. Intrinsic Neural Fields: Learning Functions on Manifolds. arXiv:2203.07967 [cs.CV] https://arxiv.org/abs/2203.07967

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2024. Neural operator: learning maps between function spaces with applications to PDEs. *J. Mach. Learn. Res.* 24, 1, Article 89 (March 2024), 97 pages.

Artiom Kovnatsky, Michael Bronstein, Alex Bronstein, Klaus Glashoff, and Ron Kimmel. 2013. Coupled quasi-harmonic bases. *Computer Graphics Forum* 32, 2pt4 (2013), 439–448.

Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. 2021. DPC: Unsupervised Deep Point Correspondence via Cross and Self Construction. In *Proceedings of the International Conference on 3D Vision (3DV)*. 1442–1451.

B. Levy. 2006. Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. 13–13.

Lei Li, Nicolas Donati, and Maks Ovsjanikov. 2022. Learning Multi-resolution Functional Maps with Spectral Attention for Robust Shape Matching. arXiv:2210.06373 [cs.CV] https://arxiv.org/abs/2210.06373

Tatsuya Harada Li, Yang. 2022. Non-rigid Point Cloud Registration with Neural Deformation Pyramid. arXiv:2205.12796

Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. 2021. Neural Scene Flow Prior. arXiv:2111.01253 [cs.CV] https://arxiv.org/abs/2111.01253

Yang Li and Tatsuya Harada. 2022. Lepard: Learning partial point cloud matching in rigid and deformable scenes. arXiv:2111.12591 [cs.CV] https://arxiv.org/abs/2111.12591

Or Litany, Tal Remez, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. 2017. Deep Functional Maps: Structured Prediction for Dense Shape Correspondence. 5660–5668. doi:10.1109/ICCV.2017.603

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *TOG* 34, 6 (2015), 248:1–248:16.

Filippo Maggioli, Daniele Baieri, Emanuele Rodolà, and Simone Melzi. 2025. Rematching: Low-resolution representations for scalable shape correspondence. In *European Conference on Computer Vision*. Springer, 183–200.

Robin Magnet, Kevin Bloch, Maxime Taverne, Simone Melzi, Maya Geoffroy, Roman H Khonsari, and Maks Ovsjanikov. 2023. Assessing craniofacial growth and form without landmarks: A new automatic approach based on spectral methods. *Journal of Morphology* 284, 8 (2023), e21609.

Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. 2022. Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization. In *2022 International Conference on 3D Vision (3DV)*. IEEE. doi:10.1109/3dv57658.2022.00061

Riccardo Marin, Enric Corona, and Gerard Pons-Moll. 2024. NICP: Neural ICP for 3D Human Registration at Scale. arXiv:2312.14024 [cs.CV] https://arxiv.org/abs/2312.14024

Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. 2020. Correspondence Learning via Linearly-invariant Embedding. arXiv:2010.13136 [cs.CV]

Simone Melzi. 2019. Sparse representation of step functions on manifolds. *Computers & Graphics* 82 (2019), 117–128.

Simone Melzi, Riccardo Marin, Pietro Musoni, Filippo Bardon, Marco Tarini, and Umberto Castellani. 2020. Intrinsic/extrinsic embedding for functional remeshing of 3D shapes. *Computers & Graphics* 88 (2020), 1–12.

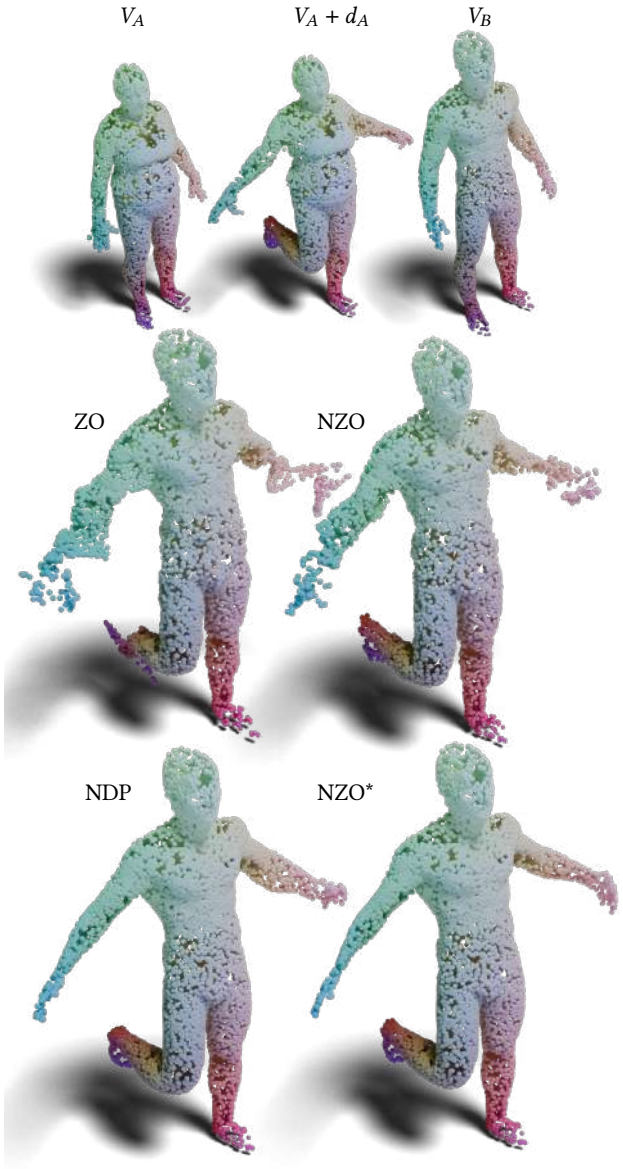

Fig. 10. Neural deformation transfer between non-isometric shapes. For ZO and NZO, the deformation field is defined on the LBO, and for NDP and NZO* it is defined on the RFF. Intrinsic approaches fail to transfer good deformations from one shape to the other, while considering extrinsic representations, as in NZO* and NDP, we can get near optimal field transfer.

Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2019a. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association.

Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. 2019b. ZoomOut: Spectral Upsampling for Efficient Shape Correspondence. *ACM Transactions on Graphics (TOG)* 38, 6, Article 155 (Nov. 2019), 14 pages.

Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. 2018. Localized Manifold Harmonics for Spectral Shape Analysis. *Computer Graphics Forum* 37, 6 (2018), 20–34.

Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael Bronstein, and Maks Ovsjanikov. 2018. Improved Functional Mappings via Product Preservation. *Computer Graphics Forum* 37, 2 (2018), 179–190.

Dorian Nogneng and Maks Ovsjanikov. 2017. Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum* 36, 2 (2017), 259–267.

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. 2012. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30:1–30:11.

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. 2016. Computing and Processing Correspondences with Functional Maps. In *SIGGRAPH ASIA 2016 Courses*. ACM, New York, NY, USA, Article 9, 60 pages.

G. Pai, J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. 2021. Fast Sinkhorn Filters: Using Matrix Scaling for Non-Rigid Shape Correspondence with Functional Maps. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 384–393.

Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. *ICCV* (2021).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703

Ulrich Pinkall and Konrad Polthier. 1993. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2, 1 (1993), 15–36.

Alessandro Raganato, Gabriella Pasi, and Simone Melzi. 2023. Attention And Positional Encoding Are (Almost) All You Need For Shape Matching. *Computer Graphics Forum* (2023). doi:10.1111/cgf.14912

Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. 2020. MapTree: Recovering Multiple Solutions in the Space of Maps. *ACM Trans. Graph.* 39, 6, Article 264 (nov 2020), 17 pages.

Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. 2021. Discrete Optimization for Shape Matching. *Computer Graphics Forum* 40, 5 (2021), 81–96.

Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. 2019. Structured regularization of functional map computations. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 39–53.

Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. 2018. Continuous and Orientation-preserving Correspondences via Functional Maps. *ACM Transactions on Graphics (TOG)* 37, 6 (2018).

Alessandro Riva, Alessandro Raganato, and Simone Melzi. 2024. Localized Gaussians as Self-Attention Weights for Point Clouds Correspondence. arXiv:2409.13291 [cs.CV]

Emanuele Rodolà, Michael Möller, and Daniel Cremers. 2017. Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 700–711.

Yusuf Sahillioğlu. 2020. Recent advances in shape correspondence. *The Visual Computer* 36, 8 (2020), 1705–1721.

Nicholas Sharp and Keenan Crane. 2020. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum* 39, 5 (2020), 69–80. doi:10.1111/cgf.14069

Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*. 109–116.

Robert W Sumner and Jovan Popović. 2004. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, Vol. 23. ACM, 399–405.

Mingze Sun, Shiwei Mao, Puhua Jiang, Maks Ovsjanikov, and Ruqi Huang. 2023. Spatially and Spectrally Consistent Deep Functional Maps. arXiv:2308.08871 [cs.CV] https://arxiv.org/abs/2308.08871

Ramana Sundararaman, Nicolas Donati, Simone Melzi, Etienne Corman, and Maks Ovsjanikov. 2024. Deformation Recovery: Localized Learning for Detail-Preserving Deformations. *ACM Transactions on Graphics* 43, 6 (2024). doi:10.1145/3687968

Ramana Sundararaman, Riccardo Marin, Emanuele Rodola, and Maks Ovsjanikov. 2022. Reduced Representation of Deformation Fields for Effective Non-rigid Shape Matching. *Advances in Neural Information Processing Systems* 35 (2022).

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. arXiv:2006.10739 [cs.CV]

Jiapeng Tang, Lev Markhasin, Bi Wang, Justus Thies, and Matthias Nießner. 2023. Neural Shape Deformation Priors. arXiv:2210.05616 [cs.CV] https://arxiv.org/abs/2210.05616

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2020. Deep Image Prior. *International Journal of Computer Vision* 128, 7 (March 2020), 1867–1888. doi:10.1007/s11263-020-01303-4

Giulio Viganò and Simone Melzi. 2024. Bijective upsampling and learned embedding for point clouds correspondences. *Computers and Graphics* 122 (2024), 103985. doi:10.1016/j.cag.2024.103985

Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 2017. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*. IEEE, Piscataway, NJ, USA, 5524–5532.