

Attention Based Optimization for 3D Shape Registration

A. Riva¹ , L. Olearo¹  and S. Melzi¹ 

¹University of Milano-Bicocca, Italy

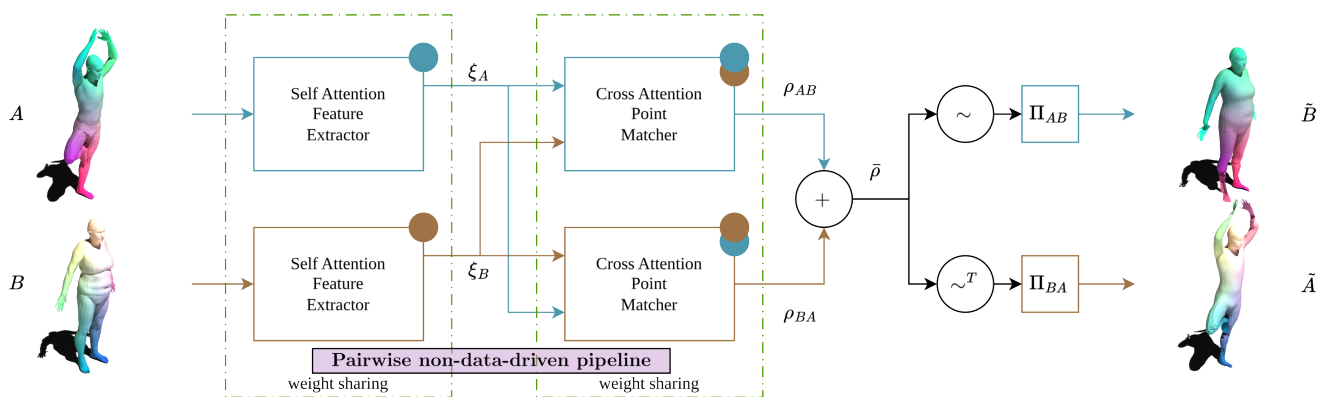


Figure 1: A visualization of our architecture, from left to right, the two input shapes are processed by the self attention feature extractor (Section 4.2), producing ξ_A and ξ_B , then the cross attention point matcher is applied in both directions (Section 4.3), obtaining ρ_{AB} and ρ_{BA} , that are subsequently composed into $\bar{\rho}$ and, finally, in Π_{AB} and Π_{BA} (Section 4.4). The brown and blue colors represent the shape the modules refer to as input sequence, thus determining the number of points (tokens) in the sequence, in blue we represent A and in brown B. In the cross-attention layers, we have balls of both colors; the one in front represents the context, while the one behind represents the query. With $\sim$ we denote the softmax function.

Abstract

Transformers are sequence-to-sequence architectures originally designed to handle structurally rigid and order-sensitive data, such as text and images. At their core, they exploit the attention mechanism, which is permutation-equivariant and relies on computing token-to-token relationships. These models have been applied to 3D geometry in several instances, achieving discrete success across tasks such as shape generation, segmentation, classification, shape matching, and registration. While existing 3D geometry methods use transformers as traditional learners, we present an approach that reinterprets the transformer as an optimization pipeline for shape correspondence. By fitting the model directly to a shape pair, our method eliminates the need for large training datasets, providing a category-agnostic solution. In particular, we focus on the use of attention weights, tailored to encode token-to-token information, to inject and extract point-to-point information during the processing of one or more meshes. We demonstrate, that self- and cross-attention mechanisms can, by design, serve as feature extractors and matching solvers, respectively. Furthermore, instead of deriving correspondence from the final output of the network, we exploit the cross-attention weights directly as the permutation matrix. This framework not only achieves robust shape matching and registration but also provides a theoretically grounded, interpretable approach to attention for unstructured 3D data. Notably, our work represents an approach that leverages the transformer architecture as an end-to-end pipeline for shape correspondence, operating effectively without requiring additional training data.

CCS Concepts

• **Computing methodologies** → *Shape analysis*; • **Theory of computation** → *Computational geometry*;

1. Introduction

Shape correspondence is a fundamental task in geometry processing, and it is essential for various downstream tasks, such as statistical shape analysis, texture transfer, and shape parametrization, among others. This problem has different possible formulations and thus admits several approaches, including matching a discrete set of points, rigid or non-rigid registration, or fitting. Methods for shape correspondence range from pure combinatorial methods to deep-learning-based models.

The general pipeline for point-to-point shape correspondence typically involves three main stages: (i) feature extraction, where each point is represented by a high-dimensional embedding robust to expected shape deformations, obtained either through geometry-based descriptors or learned representations; (ii) feature comparison, in which similarities between point embeddings are measured according to a chosen metric; and (iii) correspondence estimation or refinement, where initial matches are refined to produce a consistent point-to-point mapping or registration.

The transformer architecture [VSP*17] has rapidly emerged as one of the most widely adopted deep-learning models today. This impressive impact arises from its peculiarities and its central role in large language models (LLMs) that are indeed reshaping AI and our lives. First developed for machine translation [VSP*17], the transformer has rapidly generalized beyond language to domains ranging from chemistry [S*19] to computer vision [LLD*21], just to name a few. Transformers also impacted the fields of computer graphics and geometry processing. Strongly related to our work, the method by Trappolini et al. [TCM*21] adopts a pointwise tokenization of the input and uses a transformer-based pipeline for non-rigid 3D pointcloud registration. Subsequent works [RPM23, RRM24] analyzed purely transformer-encoder architectures for shape matching tasks and the information learned in attention weights. These studies demonstrate promising outcomes, high-quality dense correspondences, reduced supervision, improved generalization. At the same time, some limitations have emerged:

- **Data dependence:** these models typically require large-scale datasets with dense ground-truth correspondences for training, which limits their usability in settings where annotated data are scarce or costly to acquire.
- **Generalization to unseen deformations:** current transformer-based models may exhibit limited robustness when confronted with shape categories or deformation regimes not represented in the training data, leading to degraded correspondence accuracy.
- **Computational scalability:** the quadratic complexity of full self-attention with pointwise tokenization severely limits scalability to high-resolution or very large pointclouds, making both training and inference computationally expensive and memory-intensive.

This work addresses discrete matching and non-rigid registration by utilizing a minimal set of landmarks (~ 5) to ground the initial point representations with semantic information. To demonstrate the model's flexibility and potential for full autonomy, we further evaluate the pipeline using input features derived from a standard neural backbone [SACO22] and a training-based representation [MSP*23].

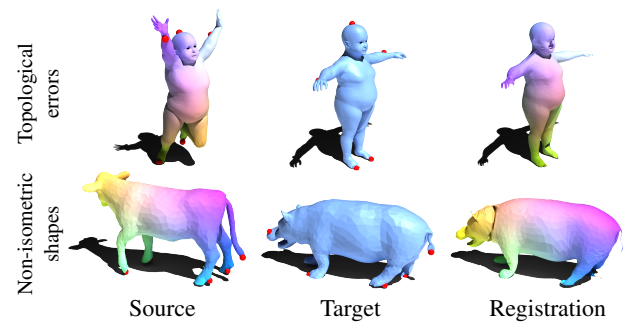


Figure 2: Qualitative results of our registration pipeline demonstrating accurate and smooth shape correspondences across different classes and deformation, including cases with topological noise (top row) and strongly non-isometric pairs (bottom row). From left to right: the source shape with the landmarks in red, the original target shape in blue with the landmarks in red, our registration output with color coding representing corresponding points.

As depicted in Figure 1, the proposed solution focuses on the role of the transformer architecture *as a pipeline* for the shape correspondence problem, inspired by the analogy between machine translation (mapping a source sentence in language A to a target sentence in language B) and shape correspondence (mapping points from a source shape A to points on a target shape B). In this light, the self-attention layers act as a *feature extractor*, encoding each point of the shapes into a high-dimensional representation, while the cross-attention layers act as a *matcher*, comparing representations of points from one shape to those of the other and providing a natural mechanism for estimating the correspondence. By leveraging pointwise tokenization and fostering the attention mechanism to discover geometric relations among tokens, our pipeline effectively targets diverse settings even in the presence of strong non-isometries and topological noise, as illustrated in Figure 2. We highlight that all the qualitative visualizations we report in this paper are registration results.

We can summarize our contributions as follows:

- We propose a pairwise fitting method that revisits the transformer architecture in a non-ML related fashion as a model to solve the correspondence between a pair of shapes, *without* requiring large quantities of training data with available ground-truth correspondence, but rather by performing a single instance optimization, also making it agnostic to the class of shapes and robust across different types of deformations.
- We leverage the attention mechanism to encode point-to-point relations and we exploit the cross-attention weights *directly* as correspondence maps, encoding the solution to the shape correspondence problem.
- By introducing a sampling strategy that selects not only mesh vertices but also points inside the triangles during the optimization, our model becomes explicitly informed about the continuous nature of the underlying surfaces, resulting in robustness to discretization changes and sampling variability at test time.

2. Related Works

2.1. Shape correspondence

Shape correspondence can, in general, be defined as the problem of finding correspondences between points of two surfaces. Among the possible variations of this problem, this paper focuses on non-rigid shape registration and discrete point matching. These two intertwined problems have been thoroughly explored in the literature, as they form the basis for many geometry processing applications. On one side, registration is the process of computing the transform that deforms one surface into another. Matching, on the other hand, is the task of computing a discrete map that relates points of two or more discretizations. These problems have been addressed in several ways.

Some methods employ a combinatorial approach to solve the quadratic assignment problem (QAP) [PRW94] efficiently. In [VLB*17], the authors consider an alternative linearization of the problem. More recently, in [HLC20], simulated annealing was applied to shape matching within a QAP-based method, and [RSCB22] proposes a new primal heuristic coupled with a Lagrange dual problem to achieve a faster solver that allows for much broader application.

Other approaches aim to compute local and global descriptors and subsequently perform matching in a more semantically meaningful descriptor space. In [SOG09, ASC11], the authors rely on diffusion geometry to compute signatures to match shapes. Similarly, in [MOR*18], the authors propose an eigendecomposition-free solution based on discrete time evolution and geodesic distances rather than on the Laplace-Beltrami operator. In [TSDS10], a method that exploits extrinsic geometry and leverages both signature- and histogram-based encoding is presented.

Correspondence representation methods aim to compute point correspondences indirectly. In [KLF11], the authors perform a blending of different mappings to achieve low distortions for pairs with large deformations. A milestone in this setting is the functional maps framework [OBCS*12] that concisely encodes the correspondence as a matrix of coefficients in a functional-based interpretation. This solution gave rise to a plethora of methods [PKO22, PRM*21, RMWO21, HRWO20, MRR*19, NMR*18, NO17] that, in different ways, improve the efficiency or the accuracy of the estimated solution. In [VLR*17], the authors proposed a method for smooth bijective correspondence that uses an approach based on kernel density estimation, which does not assume near-isometry.

Finally, in recent years, many learning-based solutions have been proposed, both with data-driven methods as in [TCM*21, RPM23] where transformers are applied to the matching problem, or [MRMO20, DCO22, CRB23] where the authors apply deep learning techniques to functional maps, and with neural models based optimization solutions that do not require large amounts of data as in [LH22]. Recently, in Neural Adjoint Maps (NAM) [VOM25] a neural operator is paired to a functional map and applied to an input correspondence (e.g. one computed with a standard functional maps). Similar to our method, which in this light can be interpreted as a step forward from NAM, this solution leverages a neural operator that is pairwise optimized, but with the dif-

ference that Nam only acts as a refinement and requires as input an initial correspondence.

2.2. Transformers

Transformers, first introduced in [VSP*17], are a sequence-to-sequence architecture originally developed for natural language processing. These architectures achieved state-of-the-art performance in both analysis tasks, such as language understanding, and generative tasks, such as machine translation, paving the way for much larger pre-trained models. Subsequent works leveraged the transformer paradigm to scale to billions of parameters, leading to large language models capable of few-shot and zero-shot reasoning [BMR*20, R*19], and establishing transformers as a foundational architecture across NLP and beyond.

Transformers everywhere Initially introduced for natural-language processing in tasks such as machine translation [VSP*17], the transformer has proven game-changing across a broad spectrum of domains — from biology (e.g., protein modelling [J*21b], genomics [J*21a]) and medicine (e.g., multimodal clinical inference [H*23]) to chemistry (e.g., molecular graph prediction [R*20]), software engineering (e.g., code generation and program synthesis [L*22]) and computer vision (e.g., image classification [Dos20, JGB*21], and video understanding [ADH*21]). In the fields of computer graphics and geometry processing, the same paradigm is increasingly leveraged to model 3D scenes, synthesize shapes, and reason over geometric structures [MSP*23, ZGJM21, SKHL24].

Transformers for shape analysis Transformers have also been applied to 3D shapes. In this geometric setting, a crucial aspect is the tokenization, which determines how continuous geometric data is discretized for the attention mechanism. Beyond basic pointwise tokenization [ZJJ*21], recent research explores patch-based strategies [PWT*22] to capture local-to-global structures efficiently for tasks such as semantic segmentation and object classification. The Point Transformer series makes a substantial contribution to pointwise tokenization: PTv1 [ZJJ*21] introduced vector self-attention for local features; PTv2 [WLJ*22] enhanced spatial grouping via grouped vector attention; and PTv3 [WJW*24] optimized performance through serialized point processing and large-window attention. Recent advancements further refine these architectures, such as LitePT [YRW*26], which achieves higher efficiency by streamlining sparse convolutional integrations, and Point Mamba [LZX*24], which introduces linear-complexity state-space modeling as an alternative to quadratic attention.

In parallel with analysis, generative models have transitioned to Transformer backbones to handle complex 3D distributions. Recent methods exploit Diffusion Transformers (DiT) [LLP*25] to synthesize high-fidelity meshes and pointclouds. Most recently, MeshRipple [LLG*26] further advances artist-mesh synthesis by introducing a frontier-aware connectivity-based tokenization and sparse-attention global memory.

Transformers for shape correspondence Transformer architectures have been applied to 3D data in a number of instances. In

particular, in [TCM*21], a Perceiver transformer [JGB*21] was applied for the first time to solve shape registration and matching by directly estimating the registration of an input connectivity to a target geometry. [TCM*21] inspired other works, such as [HDZ*23, HYW*24], which mainly introduced new architectural designs. In [RPM23] a lighter architecture is shown to achieve equal results leveraging an encoder only transformer with the support of Rotary Position Embeddings (RoPE) [SAL*24] to allow for a concatenation of the input sequences in a single longer sequence. This work greatly reduced the size of the model to perform the same task and showed a specialization of the heads in producing localized attention patterns similar to those observed in models trained on text data. Following up on these findings, in [RRM24], the authors explore a simplified architecture that replaces parts of the attention functions with precomputed information over the 3D shapes. By leveraging pointwise tokenization and fostering the attention mechanism to discover intrinsic relations among tokens, our pipeline targets diverse settings by treating the transformer not as a traditional learner, but as a specialized optimization solver.

3. Background

In this Section, we provide the necessary background and references to introduce the transformer architecture and the shape correspondence task.

3.1. The transformer architecture

Transformers are a sequence-to-sequence architecture originally introduced for natural language processing and later applied to a wide variety of tasks with positive results. The great success of these architectures led to an ever-growing number of variations, with different definitions of the attention mechanism that lies at the core of the transformers and new positional encodings that allow the architecture to have the notion of the order of the tokens in the otherwise permutation equivariant attention.

3.1.1. The dot-product attention

Given two sequences A and B , with $A \in \mathbb{R}^{n_A \times d_\xi}$ and $B \in \mathbb{R}^{n_B \times d_\xi}$, the multi-head attention weights for the j -th head at the i -th layer in a h -headed transformer are defined as follows:

$$\rho_{i,j} = \frac{Q_{i,j} K_{i,j}^\top}{\sqrt{d_{Att}}} \quad (1)$$

$$\Pi_{i,j} = \text{Softmax}(\rho_{i,j}) + \Pi_{i-1,j} \quad (2)$$

$$Att_{i,j} = \Pi_{i,j} V_{i,j} \quad (3)$$

Where Q , K and V are computed as:

$$Q_{i,j} = \begin{cases} AW_{i,j}^Q & i = 0 \\ \widetilde{Att}_{i-1} W_{i,j}^Q & i \neq 0 \end{cases}$$

$$K_{i,j} = BW_{i,j}^K$$

$$V_{i,j} = BW_{i,j}^V$$

With $W_{i,j}^Q$, $W_{i,j}^K$ and $W_{i,j}^V \in \mathbb{R}^{d_\xi \times d_{Att}}$, $d_{Att} = d_\xi/h$, and Att_i the concatenation of $Att_{i,j} \forall j$, ρ will indicate correlation matrices, and Π

will indicate permutation matrices. $\widetilde{Att}_{i-1}$ is the output of the preceding attention layer, that, in addition to the attention function, consists of normalization and fully connected layers. The output of the attention is first fed to a linear layer Lin_i with dimensionality d_ξ that allows for inter-head information flow. Then follows a residual connection with the input sequence, a normalization layer and a fully connected layer FF_i that typically expands and shrinks back the dimensionality of the embedding. Finally, a residual connection from the input of FF_i and a second normalization layer are applied.

3.1.2. Positional encoding

The attention mechanism is, by definition, permutation equivariant. When dealing with strictly sequential data however, like in text or images, where changing the order of the tokens can drastically change the information that is being conveyed, this permutation equivariance is detrimental to the results. To address this issue, common transformers implement a positional encoding that is in some way attached to the input sequence or injected into the attention mechanism itself. In this way, the models can learn to differentiate between different tokens and also the same tokens in different positions.

3.2. Shape correspondence notation

We model a 3D Shape as a compact and smooth surface $\mathcal{A}$ embedded in $\mathbb{R}^3$. In the discrete setting, we model $\mathcal{A}$ as a triangular mesh $M_{\mathcal{A}} = \langle A, E_A, F_A \rangle$ where A is the set of points in 3D space, E_A is the collection of edges, and F_A is the set of faces of the mesh. Given two surfaces, $\mathcal{A}$ and $\mathcal{B}$, and their discretizations $M_{\mathcal{A}}$ and $M_{\mathcal{B}}$, we define the correspondence problem as that of computing a matrix (a permutation matrix in the case of 1 : 1 correspondence) Π_{AB} such that $\bar{B} = \Pi_{AB} B$ is the registration of the geometry of $\mathcal{B}$ with the discretization of $\mathcal{A}$. In other words, $\langle \bar{B}, E_A, F_A \rangle$ is the mesh that originally represented $\mathcal{A}$ fitted on the geometry of $\mathcal{B}$. Although we adopt a notation that fits our method, the explicitly computes Π_{BA} , estimating the correspondence matrix is not mandatory. In the transformer-based solution for shape registration proposed in [TCM*21], the model was trained to directly output a new set of points $\bar{B}$ without computing the matrix.

We denote with L_A and L_B , namely the landmarks of $\mathcal{A}$ and $\mathcal{B}$, a set of ordered indices in the list of points in A and B that we know are corresponding points with respect to a ground truth map. Throughout the paper, we will use the letter n to indicate the number of elements in a sequence or the number of features. At the same time, d will denote the dimensionality of the embedding spaces.

4. Methodology

We introduce a transformer-based architecture for solving registration and shape-matching tasks in a bidirectional manner, with a single instance optimization framework and without requiring the extensive training of machine learning methods. The full implementation of our pipeline is available at the repository github.com/ariva00/Attention4ShapeRegistration. Following the structure of a classical matching pipeline, this architecture is composed of two main elements as shown in Figure 1:

- **Self Attention Feature Extractor.** This first component is a self attention transformer that acts as a pointwise feature extractor, mapping each point in the input pointcloud into a higher dimensional embedding space. This module is applied to both the shapes, representing them in the same space.
- **Cross Attention Matcher.** This component is a cross attention transformer where one shape acts as the input sequence and the other shape acts as the context sequence. This transformer is composed of multiple multi-head layers, with the exception of the final layer that needs to be single-headed as the output of this last layer is not the standard dot product attention, instead it only outputs the ρ matrix, which will serve as a starting point for the permutation matrix that solves the correspondence. This module is also applied twice, once with the $\mathcal{A}$ as input and $\mathcal{B}$ as context, and once with the roles flipped.

It is possible to relate the main components of a matching pipeline described in Section 1 (feature extraction, feature comparison, and correspondence estimation) to the blocks illustrated in Figure 1. Our architecture relies on attention layers only, with no positional encoding, this reflects the nature of the data, that does not present any natural order relationship, providing a permutation equivariant solution. Additionally, we fit a model for each couple of shapes starting from a set of landmarks, relieving the need for an extensive and diverse dataset that affects deep learning methods such as [TCM*21] and [RRM24]. Finally, our architecture is capable of handling new points sampled on the surface without the need for any additional training. We define an architecture that outputs a permutation matrix Π_{BA} that maps the points of B over the shape $\mathcal{A}$ and vice-versa a matrix Π_{AB} that maps A over the shape $\mathcal{B}$, solving the registration in both directions simultaneously.

4.1. Data Vectors

$$\tilde{A} = \begin{bmatrix} A & D_{lmk} & \text{FourierFeat}(\langle A, D_{lmk} \rangle, n_{ff}) \end{bmatrix}$$

[c] [3] [n_{lmk}] [(3 + n_{lmk})n_{ff}]

Before the transformer pipeline the input sequences A and B are enriched with some additional information. Firstly, for each point, the geodesic distance from all the landmarks, represented by D_{lmk} , is appended to the coordinates vector, creating a longer data vector. Additionally, we concatenate to this vector the random Fourier feature transform of itself [RR07]. For each point the resulting data vector in $\tilde{A}$ and $\tilde{B}$ is a vector of c elements: 3 coordinate components, n_{lmk} distances from landmarks, and $(3 + n_{lmk})n_{ff}$ Fourier features applied to the previous elements, with n_{ff} being the number of random Fourier features frequencies. In the inset, we visualize an example of a data vector $\tilde{A}$ for a surface $\mathcal{A}$, with the dimensionality of each block reported at the bottom. Having already included the coordinates, we do not add any positional encoding to the data vectors.

While in this work we focus on the landmark guided setting, in addition to the landmark based data vectors, we also test as input to our model the output of two feature extractors, namely, DiffusionNet [SACO22] randomly initialized and optimized alongside our model, and pretrained Diff3f [DMM24]. These models are selected in order to provide a landmark-free and a learned alternative.

4.2. Self Attention Feature Extractor

Given a mesh discretization $M_X = \langle X, E_X, F_X \rangle$ and its data vectors matrix $\tilde{X} \in \mathbb{R}^{n_X \times c}$ this module outputs a matrix $\xi_X \in \mathbb{R}^{n_X \times d_\xi}$,

where n is the number of vertices of the mesh and d is the dimension of the embedding space where the transformer architecture operates. This feature extractor is a self attention multi-head transformer composed of l_{self} layers with h heads each.

4.3. Cross Attention Matcher

This cross attention transformer receives as input the embedding of the two shapes ξ_A and ξ_B coupled with the respective mesh discretizations of the surfaces and outputs a matrix $\bar{\rho}$ that represents the correspondences. The first $l_{cross} - 1$ layers of the transform are multi-head attention layers with h heads each. The final layer is a single head attention layer where only Q and K are computed and the attention weights ρ are returned as output. This module is applied twice, switching the two input sequences, producing the two output matrices ρ_{AB} and $\rho_{BA}^\top$. The $\bar{\rho}$ matrix, that relies on the bidirectionality of the task, is subsequently computed as:

$$\bar{\rho} = (\rho_{AB} + \rho_{BA}^\top) / 2. \quad (4)$$

4.4. Mapping

This matrix and its transposed are then used to perform registration and matching of points. In particular, the permutation matrices Π_{BA} and Π_{AB} are computed as follows:

$$\Pi_{AB} = \text{Softmax}(\bar{\rho}) \quad (5)$$

$$\Pi_{BA} = \text{Softmax}(\bar{\rho}^\top) \quad (6)$$

The registration mapping of A over the surface $\mathcal{B}$ is computed as $\bar{B} = \Pi_{AB}B$ and, vice-versa, $\bar{A} = \Pi_{BA}A$ is the registration of the points in B over the surface of $\mathcal{A}$. Similarly, the matching mapping of A over the surface $\mathcal{B}$ is computed for each point q as $\hat{B} = \{\argmin_{p \in B} (\bar{q} - p) | \bar{q} \in \bar{B}\}$ and conversely as $\hat{A} = \{\argmin_{q \in A} (\bar{p} - q) | \bar{p} \in \bar{A}\}$ for the mapping of B over the surface of $\mathcal{B}$.

4.5. Optimization

Although we perform all the optimization on a single pair of shapes and we do not perform any data-driven training, as we build our pipeline on the transformer architecture we adopt the terminology used in common ML settings (e.g. training, pretraining, loss, epoch). To perform the optimization we define three losses, some are used during both training and pretraining, while others are limited to one of the two phases. It is important to notice that all the optimization is performed on a single couple, as ours is not a data-driven method. Consequently, one epoch of the optimization consists in a single forward and backward step on the model.

Matching Loss Given two lists of indices representing the known correspondences L_A and L_B , we define this loss as:

$$\text{Loss}_m = \|(\Pi_{BA}A|_{L_B}) - (A|_{L_A})\| + \|(\Pi_{AB}B|_{L_A}) - (B|_{L_B})\| \quad (7)$$

Double Permutation Loss This loss is based on the bidirectionality of the matching, thus requiring that $\Pi_{AB}\bar{B} = A$.

$$\text{Loss}_p = \|\Pi_{AB}\Pi_{BA}A - A\| + \|\Pi_{BA}\Pi_{AB}B - B\| \quad (8)$$

Chamfer Loss This is the standard bidirectional chamfer loss used in common registration tasks, requiring good coverage of the points in both the direction of the matching.

$$Loss_c = \sum_{p \in B} \min_{q \in A} (\|p - q\|) + \sum_{q \in A} \min_{p \in B} (\|p - q\|) \quad (9)$$

4.5.1. Pretraining

In the pretraining phase the weights of the Cross Attention Matcher are frozen in order to train only the Self Attention Feature Extractor. At each epoch of pretraining the whole model is run to solve the correspondences between $\mathcal{A}$ and a random rotation of itself and between $\mathcal{B}$ and a random rotation of itself. This means that during the pretraining every point of the mesh is a landmark, as we know the correspondences by construction. In this phase, the only active loss is the Matching Loss defined as in 7 where A and B are the same collection of points.

4.5.2. Training

In the training phase, the model is run with $\mathcal{A}$ and $\mathcal{B}$ fixed as first and second input. At this stage all the defined losses are active, with the matching loss being computed solely on the known matches as described by the landmarks.

4.6. Resampling

Performing gradient descent in a transformer requires large quantities of memory due to the quadratic scaling of the attention function with respect to the length of the sequences. To perform the optimization on dense shapes it is possible to perform a resampling of the shape. In particular we perform a new resampling of the shape at every epoch by extracting half of the resampled points from the faces of the shape and the other half from the original set of points.

In general, $\bar{\rho}$ scales quadratically with respect to the number of points in the shape, posing a hard limit on the direct applicability of the method to larger meshes. In order to compute the registration for couple of large shapes, we define two strategies corresponding to two degrees of computational limits: on to deal with couples where $\bar{\rho}$ is too large for GPU memory and one for couples where it does not fit in CPU memory either.

4.6.1. Out of GPU memory

Optimization At training time, we perform a resampling of the points. Specifically, as illustrated in Figure 3, we sample 50% of the points from the original set of points and the remaining 50% from random positions on the mesh faces, ensuring that all landmarks are sampled. In general, a new sampling $\hat{A}$ of the points in A is defined as $\hat{A} = S_A A$ where S_A is the barycentric matrix of the sampling and we use this same matrix to also resample the matrix of landmarks distances D_{lmk} as $\hat{D}_{lmk} = S_A D_{lmk}$.

Test At test time, in order to compute $\bar{\rho} \in \mathbb{R}^{n_A \times n_B}$ we define a larger matrix $\tilde{\rho} \in \mathbb{R}^{n_S \times n_S}$, which is composed of $t \times t$ partial matrices $\tilde{\rho}_{a,b} \in \mathbb{R}^{n_S \times n_S}$, where n_S is the number of samples used to compute the partial matrices and $t = \min\{t | n_S > n_A \wedge n_S > n_B\}$. These $\tilde{\rho}_{a,b}$ matrices relate each point to multiple subsets of points and $\tilde{\rho}$ is subsequently reduced to the target $\bar{\rho}$ matrix via averaging.

Further details on the test time resampling strategy are reported in the supplementary material.

4.6.2. Out of CPU memory

In this case the number of point in the couple is so high that the $\bar{\rho}$ could not fit into CPU memory, making it impossible to directly compute it. To solve this problem we apply Rematching [MBRM24] to the input shapes and reduce the problem to the out of CPU memory case both at train time and at test time. Specifically, Rematching allows us to compute the new discretizations $M_A^{rmt} = \langle A^{rmt}, E_A^{rmt}, F_A^{rmt} \rangle$ and $M_B^{rmt} = \langle B^{rmt}, E_B^{rmt}, F_B^{rmt} \rangle$ with n_{rmt} number of vertices and the barycentric matrices S_A^{rmt} and S_B^{rmt} that interpolate functions that are defined over the new discretizations on the original meshes. Consequently, once we computed the registration $\bar{A}^{rmt}$ of B^{rmt} over the shape of A^{rmt} we compute $\bar{A} = S_B^{rmt} \bar{A}^{rmt}$ and symmetrically $\bar{B} = S_A^{rmt} \bar{B}^{rmt}$.

5. Results

5.1. Experimental Setup

In this section, we report a brief summary of the used datasets, a definition of the hyperparameters of the architecture and a definition of the evaluation metrics. We compare our method with baselines from the state of the art of landmark-based shape matching: **Functional Maps (FMaps)** [OBCS*12] with landmarks as descriptors. **ZoomOut** [MRR*19] as a refinement method for FMaps with landmarks due to its recognized success and widespread applications. **Neural Adjoint Maps (NAM)** [VOM25] as a more recent refinement method for FMaps that is less reliant on isometry. And finally, **Neural Deformation Pyramid (NDP)** [LH22] in its landmarks-based formulation as a non-spectral alternative.

The baseline methods are run using the original source code or re-implemented with the GeomFuM [VLP*] library and Pytorch.

5.1.1. Dataset

We perform evaluations on four datasets:

- **FAUST**. We use the registrations of the SMPL model [LMR*15] over the quasi-isometric human bodies scans of different subjects in different poses [BKL*16]. The evaluation is performed on the dense correspondence defined by the SMPL template.
- **TOPKIDS**. The TOPKIDS dataset [LRB*16] is composed of a template shape and 25 shapes representing the same subject in different poses with significative topological noise. The evaluation is performed by matching each shape with the template with a quasi-dense ground-truth correspondence (>80% of points).
- **SMAL-R**. The SMAL dataset [ZKJB17], remeshed as in [CRB23] to have different connectivity, has been evaluated on a quasi-dense correspondence of points. SMAL-R contains 48 quadrupeds among eight species, with very low cross-species isometries.
- **SHREC'20**. The SHREC'20 dataset [DLR*20] is a challenging, highly non-isometric dataset of quadrupeds with missing parts and topological noise. Each shape is annotated with a sparse set of points (approx. 40 points), we evaluate by taking the average on the 5 test sets using for each pair the common points without the selected landmarks.

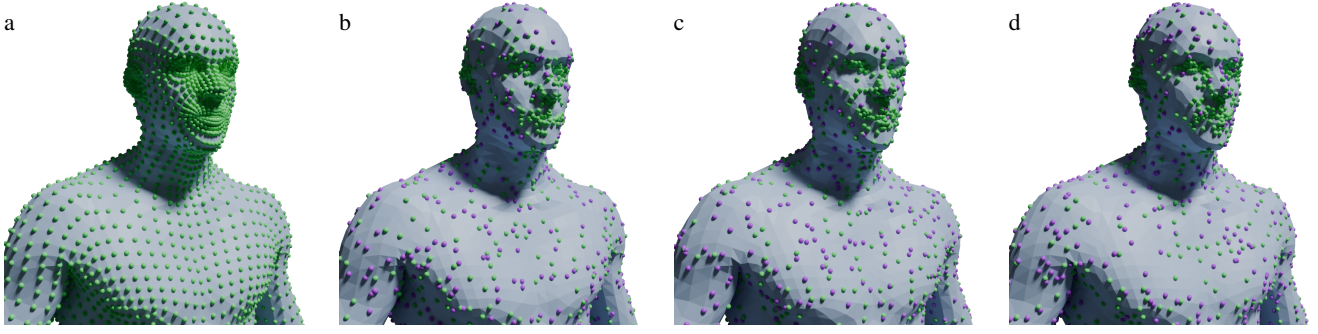


Figure 3: A visualization of the original point set (a) and the random resampling at different epochs of the optimization (b-d). In green are shown the points of the original point set, and in purple the points sampled on the faces of the mesh.

5.1.2. Architecture Hyperparameters

All the optimized models follow the same hyperparameters configurations: we selected an embedding dimension $d_{\xi} = 256$ to represent data vectors built with number of Fourier features $n_{ff} = 8$ and number of landmarks $n_{lmk} = 5$ for humanoid shapes and $n_{lmk} = 6$ for animals. The architecture is composed of $l_{self} = 4$ layers of self attention in the feature extractor and $l_{cross} = 4$ in the cross attention matcher. All the attention layers but the last are composed of $h = 8$ attention heads. To fit the model we use a new resampling of 1000 points at each step of the optimization and we perform 500 pretraining epochs followed by 1000 epochs of training.

5.1.3. Metrics

We evaluate the matching results for each couple and report in the table the average over the tested couples. The evaluation metrics are the following:

- **Geodesic Error:** for each point $p \in B$ of known correspondence in the source shape this error is computed as the geodesic distance between the ground truth matching point $q \in A$ and the point $\tilde{q} \in A$ that minimizes the distance from $\tilde{p} \in \tilde{A}$ mapping of p on the surface $\mathcal{A}$.
- **Euclidean Error:** for each point $p \in B$ of known correspondence in the source shape this error is computed as the euclidean distance between the ground truth matching point $q \in A$ and $\tilde{p} \in \tilde{A}$.
- **Dirichlet Energy:** the Dirichlet Energy is a commonly used metric in registration that represents the overall smoothness of the mapping between the surfaces.
- **Matching Dispersion:** this metric is derived from the coverage as defined in [RPWO18]. However, as we perform matching between meshes with different cardinalities we divide by the ratio of points between the source shape and the target shape.

It is important to notice that while matching dispersion and Dirichlet energy can be computed on the whole mesh, the Euclidean error and the geodesic error can be computed only on the points for which we have a ground truth correspondence, with the number of points of known correspondence varying widely across different datasets. Additionally, we compute the metrics using both the registration mapping and the point matching mapping.

Table 1: In table are reported the registration and matching results of our method compared with state of the art methods. The table is divided in accordance with the type of ground-truth correspondence, going from DENSE where we have a one-to-one correspondence between source and target, to QUASI-DENSE where most points (>80%) have a singular matching point in the target, to SPARSE with only a handful of points having a defined ground-truth match.

	Method	Euclid Err↓	Geod Err↓	Dirichlet↓	Match Disp↑
DENSE	FMaps	0.0401	0.0400	0.0016	0.3236
	ZoomOut	0.0207	0.0195	0.0009	0.6969
	NDP	0.0702	0.0886	0.0020	0.4893
	NAM	0.0245	0.0240	0.0009	0.6323
	Ours (Registration)	<u>0.0227</u>	<u>0.0234</u>	0.0007	0.5831
	Ours (Match)	0.0230	<u>0.0234</u>	0.0010	0.5831
QUASI-DENSE	FMaps	0.1051	0.1078	0.0024	0.2627
	ZoomOut	0.0815	<u>0.0835</u>	0.0040	0.6011
	NDP	0.0971	0.1019	0.0007	0.5351
	NAM	0.1244	0.1195	0.0032	0.1987
	Ours (Registration)	0.0789	0.0830	<u>0.0016</u>	<u>0.5766</u>
	Ours (Match)	<u>0.0796</u>	0.0830	0.0018	<u>0.5766</u>
	FMaps	0.0756	0.0744	0.0018	0.2162
	ZoomOut	0.0625	0.0611	0.0065	0.4706
	NDP	0.0555	0.0675	0.0010	0.4818
	NAM	0.0617	<u>0.0589</u>	0.0015	<u>0.4694</u>
SPARSE	Ours (Registration)	0.0390	0.0392	0.0007	0.5464
	Ours (Match)	<u>0.0396</u>	0.0392	<u>0.0009</u>	0.5464
	FMaps	0.1381	0.1411	0.0077	0.2037
	ZoomOut	0.1454	0.1480	0.0152	0.3495
	NDP	0.0698	0.0649	0.0013	<u>0.5016</u>
	NAM	0.1041	<u>0.0962</u>	0.0082	0.2860
SHREC'20	Ours (Registration)	<u>0.0950</u>	0.0992	<u>0.0044</u>	0.5248
	Ours (Match)	0.0952	0.0992	0.0053	0.5248
Average	FMaps	0.0897	0.0908	0.0034	0.2516
	ZoomOut	0.0775	0.0780	0.0067	<u>0.5295</u>
	NDP	0.0732	0.0807	0.0013	0.5020
	NAM	0.0787	<u>0.0747</u>	0.0035	0.3966
	Ours (Registration)	0.0589	0.0612	<u>0.0019</u>	0.5577
	Ours (Match)	<u>0.0594</u>	0.0612	0.0023	0.5577

5.2. Quantitative Evaluation

The evaluation results in Table 1 show the performances of our method compared to other shape correspondence, non-data-driven, landmark-based, methods. We report, for our method both the registration and the matching results. It is easily noticeable how the values for geodesic error and matching dispersion are always equal in the two versions, this is due to the fact they are computed via nearest neighbor from the registration to the target, factually performing the same operation that we use to compute the matching from the registration. On average our model achieves the best performances across datasets in terms of both Euclidean and geodesic error as well as matching dispersion, while the Dirichlet energy is not far from that of the best performing competitor, that is NDP. Concerning the single datasets, ZoomOut achieves better results than our method on the FAUST dataset thanks to the strong isometric prior of the method, nevertheless, our method is second best, also outperforming NAM. On the TOPKIDS dataset our method registers the lowest geodesic and euclidean errors while producing higher Dirichlet energy and a weaker coverage compared to NAM and ZoomOut respectively. The numerical results show our method outperforming the competitors on all the metrics on the SMAL-R dataset, where the isometry-reliant methods struggle the most due strong cross-species non-isometries. NDP and NAM outperform our method on the SHREC'20 dataset in terms of geodesic and euclidean error and of matching dispersion. It is worth noting that NAM is not much different from our results on this dataset, highlighting how NDP is really the only outstanding method on this dataset, which is not only evaluated on a sparse correspondence but also presents dataset-wise common landmarks exclusively on one side of the mesh due to some partial shapes being cut down the middle. This poorly covering landmarks might be affecting the performances of the methods, highlighting a possible sensibility to landmarks quality.

5.3. Qualitative results

In the following section and throughout the paper the renderings show registration results with the source mesh keeping its connectivity on the registered point set. In cases of topological errors in the source mesh, a correct registration can lead to artifacts in the visualizations, such as single faces connecting the hands to the feet of a human shape. In Figure 4, we visualize the registration comparison of our method with state-of-the-art methods. It is possible to see that our model produces very smooth results compared to the other methods. Two of the selected couples are very challenging, in particular for the functional-maps-based methods, due to the topological errors in the shapes, notably, the tip of the right hand of the kid in the first couple is connected to one of the ankles (a), while in the last couple the elephant has many connection points between legs, trunk, and tusks, significantly changing the genus of the shape and the geodesic distances between the landmarks (b), and the hippo is a non water-tight mesh, with a hole on the belly and under the feet (c). The results show that our method is more stable than competitors across datasets.

Figure 5 depicts some results on pairs of shapes of different classes compared to those seen in the quantitative comparison on benchmark datasets. These results highlight the wide applicability

Table 2: Quantitative comparison among different architectural configurations, on the FAUST dataset. The first three rows present results for different choices of active modules at training and test time (respectively, first and second columns) in the proposed pipeline. The last row shows the results inferred by the correlation between the input features. We denote the feature extractor as E and the matcher as M .

Train	Test	Euclid Err ↓	Geod Err ↓	Dirichlet ↓	Match Disp ↑
E + M	E + M	0.0227	0.0234	0.0007	0.5831
E + M	E	0.0238	0.0242	0.0010	0.5914
E	E	0.0795	0.0885	0.0158	0.4704
-	-	0.1701	0.1903	0.0002	0.0244

of our method, which, requiring no training dataset, can be applied to any couple of shapes. In Figure 6, we show a result in segmentation transfer on a couple of high-resolution (~2M vertices), non-isometric, and non-totally connected shapes. Both skulls are composed of disconnected bones that are labeled according to the correspondence of bones in the two animals. In the boxes, we highlight the different frequency of detail in the two shapes, the presence of many small connected components, and the strong non-isometry. The transfer result is smooth with respect to the mapped regions, showing a good registration even with highly different geometries. It is important to note that not all bones have a corresponding bone in the target shape, resulting in some label mismatches, which are experimentally visible in the regions corresponding to the casque of the chameleon.

6. Discussion and Analysis

In this section we present some ablation studies and further experiments to assess the effect of the architecture modules and features and to test some properties of the method. These tests are not carried out on all the datasets and are generally limited to a small randomly selected set of couples.

6.1. Architecture Ablation Study

In this section, we conduct an ablation study on the architecture of the model. In particular, our goal is to understand the contribution of the two main modules, the feature extractor (E) and the matcher (M), that compose our solution. Table 2 reports the results of the registration with our method, removing M or both modules (denoted as -) at test time or at train time. Specifically, in addition to the full model, we test the registration with our full model, tested using only the feature extractor, and the model trained and tested with only the feature extractor. Where the matcher M is not used, we collect the results using the correlation $\tilde{A}\tilde{B}^T$ in place of the matrix $\tilde{\rho}$. Examining the fitted models, it is particularly noteworthy to observe how the model that utilizes the cross-attention matcher during optimization and then performs feature correlation at test time, without the cross-attention module, achieves results very close to those of the full model. Removing the matcher at train time, however, produces far worse results, showing how the cross attention is improving the optimization of the feature extractor even if its final contribution to the solution is minimal.

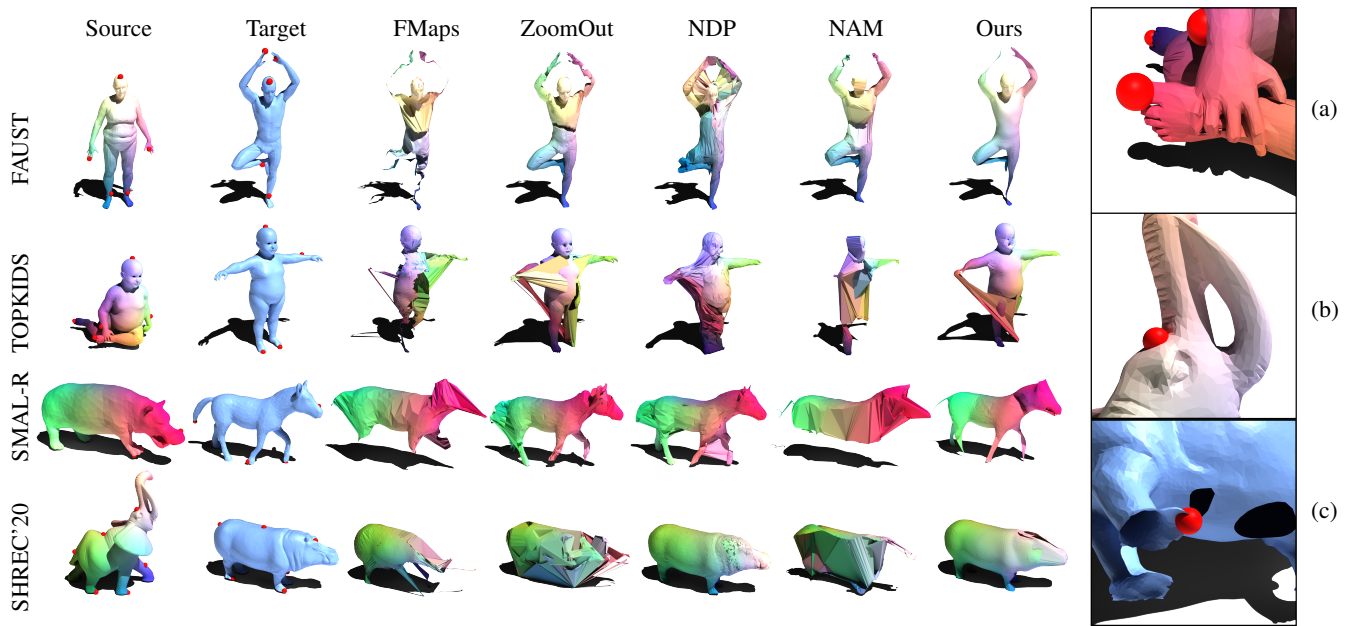


Figure 4: Qualitative results of the state-of-the-art methods compared with our method. From top to bottom are shown: the source shapes with the landmarks denoted in red, the target shapes with the corresponding landmarks, and the results of the registration of the different methods. In the boxes on the right, details of topological errors in the meshes for TOPKIDS (a), and SHREC'20 source (b) and target (c).

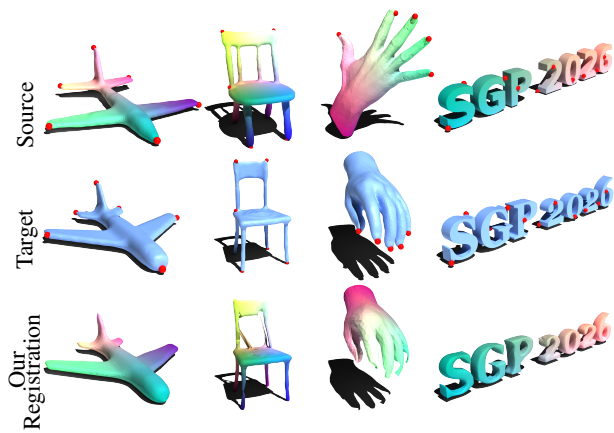


Figure 5: Qualitative results on different classes, including objects from [CGF09] and a 3D text example. The top row shows the source shapes with landmarks in red, followed by the target shapes in the second row and the registration results, demonstrating the category-agnostic applicability of our method.

6.2. Features Ablation Study

To evaluate the effectiveness of the selected features we perform an ablation study of the input data vectors by removing one component at a time and comparing the results of the different configurations. To assess the quality of the features before optimiza-

tion we also compare the registration inferred by the correlation of the input data. Additionally, we test two feature extractors to replace the landmark based data vectors: DiffusionNet [SACO22] and Diff3f [DMM24]. The DiffusionNet model is randomly initialized for each couple and optimized alongside our model, in particular it is treated as part of the feature extractor and consequently its weights are updated both at pretraining and training time. In order to be able to optimize both our model and DiffusionNet we recur to the “Out of CPU memory” strategy of resampling for the corresponding experiments. On the other hand the Diff3f model is used with frozen weights and is thus the same for every couple of shapes and does not impact much on the computational cost of the optimization.

To better understand the nature of the error of the fitted models, in addition to the geodesic error, we also report the symmetric geodesic error (i.e. for each point we consider the error from its corresponding or the symmetric point of its correspondent). For each configuration alongside the input features we report whether we used our full model ($E + M$), the matcher alone (M), or the simple correlation between vectors ($-$).

The results in Table 3 show that the input features alone are insufficient to perform a good registration and our method, with both its modules, improves the quality of the correspondence. It is noticeable how removing the landmark distances (L) from the feature vector the results are sensibly degraded, both with our model and in the correlation between the features, this is expected as the landmarks carry the semantic information that is necessary for the registration. It is however interesting to notice how, in the corre-

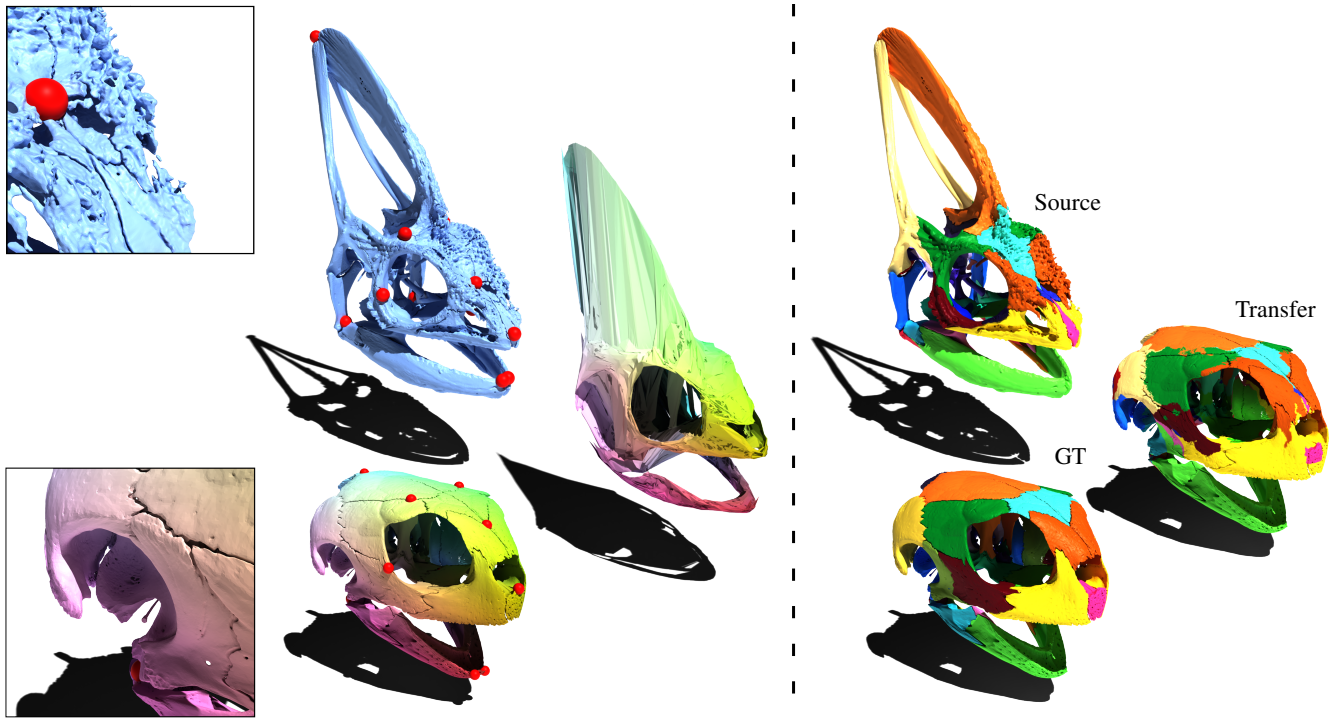


Figure 6: Qualitative results on a couple of challenging shapes and an example of application for segmentation transfer. In the boxes, closeups highlight the difference in surface smoothness and the presence of many connected components. The original meshes are made available at MorphoSource by the Florida Museum of Natural History, Division of Herpetology. Specimens: UF:Herp:191369 *Chamaeleo calyptratus* ([ark:/87602/m4/769445](https://morphoSource.org/ark:/87602/m4/769445)), UF:Herp:51413 *Chelonia mydas* ([ark:/87602/m4/615952](https://morphoSource.org/ark:/87602/m4/615952)).

lation tests removing the coordinates improves results, with the features becoming rotation invariant, but when using our architecture the best results are achieved with the full data vector (FICIL). This is due to the model learning to disambiguate points with similar landmark distance by leveraging the coordinates. Thanks to the attention, coordinate features, that carry very little information pointwise, are compared and related to those of the other points in more complex ways than simple correlation. Fitting DiffNet for each couple performs poorly both with our architecture and with straight correlation, the symmetric geodesic error however shows that most of the error is due to symmetry ambiguity, revealing the main weakness of this approach. Diff3f on the other hand performs particularly good in combination with our architecture comprising both self attentions and cross attentions. Overall, these results show that our method requires some semantic information to be able to perform matching discerning inner symmetries, be it in the landmarks or in a corpus of data, the likes of which parts of Diff3f are trained on. This is not something unexpected as it is a fairly common problem in shape matching [DCO22].

6.3. Bidirectionality

Our method strongly relies on the bidirectional nature of the matching problem, to better understand the influence of the bidirectional architecture we tested the registration on 20 random couples of the FAUST dataset applying the cross-attention matcher from Sec-

Table 3: Quantitative comparison among different feature and architectural configurations, on the FAUST dataset. The first column indicates the features used in the optimization, with *F* *C* and *L* indicating, respectively, the use of Fourier features, coordinates, and landmarks (as features). The second block of rows shows results with input features extracted by DiffusionNet (DiffNet) and by Diff3f. The second column indicates whether which parts of the architecture are used, with *M* indicating the point matcher and *E* the self attention feature extractor and finally - to represent the correlation without our modules.

	Input	Method	Euclid Err ↓	Geod Err ↓	Geod Err (Sym) ↓	Dirichlet ↓	Match Disp ↑
landmark-based	F C L	E + M	0.0195	0.0196	0.0194	0.0006	0.5836
	C L	E + M	0.0229	0.0240	0.0234	0.0006	0.4330
	F L	E + M	0.0211	0.0219	0.0218	0.0008	0.4980
	F C	E + M	0.0929	0.1075	0.0848	0.0039	0.4423
	F C L	-	0.0789	0.0849	0.0780	0.0025	0.1768
	C L	-	0.1782	0.1842	0.1763	0.0002	0.0212
	F L	-	0.0570	0.0621	0.0565	0.0008	0.1465
	F C	-	0.3064	0.3334	0.2941	0.0023	0.1561
landmark-free	DiffNet	E + M	0.1015	0.1668	0.0299	0.0025	0.4616
	DiffNet	M	0.1266	0.1899	0.0553	0.0028	0.2229
	DiffNet	-	0.1029	0.1660	0.0396	0.0035	0.2941
	Diff3f	E + M	0.0237	0.0427	0.0156	0.0011	0.6522
	Diff3f	M	0.2188	0.2755	0.2046	0.0116	0.1674
	Diff3f	-	0.2670	0.2894	0.2846	1.06E-06	0.0003

Table 4: Average of the difference between the evaluation metrics in the two directions our method solves the correspondence for.

	Euclid Err		Geod Err		Dirichlet		Match Disp	
SMAL-R	0.0038	(9.8%)	0.0040	(10.1%)	0.0002	(30.3%)	0.0099	(1.8%)
FAUST	0.0022	(9.9%)	0.0015	(6.3%)	0.0001	(9.4%)	0.0066	(1.1%)
TOPKIDS	0.0114	(14.5%)	0.0101	(12.2%)	0.0012	(75.6%)	0.0778	(13.5%)
SHREC'20	0.0161	(16.9%)	0.0171	(17.2%)	0.0042	(95.1%)	0.0871	(16.6%)

tion 4.3 only once (i.e., in a single direction), and setting $\bar{\rho} = \rho_{BA}^\top$ in Eq. 8. The registered geodesic error is 0.0288 that is significantly higher than the one reported in the first row of Table 3, that is tested on the same couples.

Furthermore, we evaluate in Table 4 the difference in the metrics for the correspondence of $\mathcal{A}$ over $\mathcal{B}$ and of $\mathcal{B}$ over $\mathcal{A}$. The difference results to be small compared to the average of the metrics as reported in Table 1, indicating also a good bidirectionality of the method. The difference on matching dispersion on the SHREC'20 and TOPKIDS datasets results to be particularly high due to the presence of partial shapes and topological errors.

6.4. Scalability

In Table 5, we report the timing results of the optimization over couples with different cardinalities. These tests are run on a desktop PC equipped with an NVIDIA RTX 3060 with 12GB of VRAM, an Intel i7-12700K CPU, and 32GB of RAM. We choose to report the results obtained on this machine, rather than on more powerful systems, for broader applicability. In this setting, the size limit for GPU memory is approximately 2500 points per shape, while the limit for CPU memory is approximately 15000 points per shape. Consequently, we resample all the tested couples, except for the first one, and perform the preprocessing and resampling from Rematching [MBRM24] on the two larger shapes. The results on the non-remeshed shapes show that the computational time per vertex is fairly constant around 0.025 seconds per vertex. The results on the remeshed shapes show that, thanks to the remeshing technique, we are able to treat shapes with more than 1000 times the number of vertices with a computation time that is less than twice as long. It is necessary to point out that the computation time for the largest couple is lower than that of the second largest due to the result of Rematching, which, fixing a target number of vertices, produces results of different cardinality depending on the input meshes. In the two tested couples, both remeshed with a target cardinality of 10000, the first shapes are remeshed at 10000 vertices each, while the second ones at 6000. This difference in the number of vertices is realistically due to the nature of the meshes, in particular, the input shapes and the results of the registration on the last couple can be seen in Figure 6, also highlighting the presence of small connected components and more complex geometries with respect to the other tested shapes, that are all extracted from common matching datasets. Taking into account the number of vertices after the remeshing, instead of the original cardinality, the time per vertex would rescale to a value around 0.025, confirming the results of the other couples.

The decision to resample the shape instead of applying Rematching in the “out of GPU memory” scenario stems from the fact that

Table 5: Computational time required for the registration of couples with different cardinalities. The computation time is reported in both seconds and as a percentage with respect to the smallest tested couple. The last two couples are the only ones where Rematching has been used, all other couples are registered using our resampling method.

$n_A + n_B$	Preprocess [s]	Optim [s]	Time [s]	Time[%]	Time/vertex [s]
2000	0.73	248.86	249.60	100.00%	0.1248
10432	8.92	298.71	307.63	123.25%	0.0295
13780	14.97	321.85	336.82	134.95%	0.0244
20002	36.55	425.77	462.32	185.23%	0.0231
112706	34.79	470.99	505.78	202.64%	0.0045
3822120	45.39	346.90	392.29	157.17%	0.0001

the resampling allows us to optimize with points lying on the original surface and still compute the registration for each point of the input as explained in Subsection 4.6. Conversely applying Rematching allows us to compute the registration only for the new point set and rely on interpolation for the original point set. While this enables our method to work with larger shapes it also introduces an error. To evaluate this we performed the same experiment reported in the first line of Table 3 but with Rematching applied before to input. The result shows a 34% increase in the geodesic error, with a nominal value of 0.0263 opposed to the 0.0196 reported without Rematching.

We also report, in Table 6, the average computational time of each method tested on the FAUST dataset. For these results a server machine equipped with an NVIDIA RTX A5000 with 24GB of VRAM, an Intel i9-10900X CPU, and 64GB of RAM has been used. While our method takes a longer time to optimize, it can be used to work with different discretizations of the same surfaces, even ones with much higher densities, without any further optimization. It is also worth noting that the current implementation of the attention is not particularly efficient, as we favored a very straightforward implementation, where we could easily work with the attention weights, instead of complex designs such as FlashAttention2 [Dao24], which could provide a significative speed up.

Table 6: Average computational time of the methods on the FAUST dataset.

Method	Avg. Time [s]
FMaps	0.7432
ZoomOut	8.4574
NDP	3.6786
NAM	31.0081
Ours	256.1043

6.5. Sampling Invariance

As explained in Subsection 4.6 the method relies on resamplings in order to work with large shapes and also to introduce sampling invariance. To test the latter we take isometric couples, that in our case is the whole test set of FAUST, and compute the barycentric sampling matrices that composes a sampling of 50% of the points randomly picked from the surface and the other half extracted from the original point set, plus the selected landmarks. This resampling allows to obtain a new dense correspondence to use as ground truth and also retains information coming directly from the original point set where the model is trained. The evaluation is then performed on the subset of points sampled on the surface. We test two techniques for the registration of the resamplings, given a resampling of the

Table 7: Results of the registration between two resampling of the input shapes in the FAUST dataset.

	Euclid Err↓	Geod Err↓	Dirichlet ↓	Match Disp ↑
Original	0.0227	0.0234	0.0007	0.5831
ρ -resampling	0.0236	0.0243	0.0019	0.5755
resampling	0.0214	0.0225	0.0135	0.6509

meshes $\hat{A}$ and $\hat{B}$, and the barycentric matrices S_A and S_B with $S_A = S_B$ such that $\hat{A} = S_A A$ and $\hat{B} = S_B B$, in ρ -resampling we compute the registration between $\hat{A}$ and $\hat{B}$ by applying the resampling to the $\hat{\rho}$ matrix as $\hat{\rho} = S_A \hat{\rho} S_B^T$ and subsequently compute $\hat{\Pi}_{AB}$ and $\hat{\Pi}_{BA}$ as in Equations 5 and 6, without requiring to apply the transformer to the new data. Conversely, in *resampling* we run a forward step on the new input set and obtain the new $\hat{\rho}$ matrix. The results in Table 7 show that the registration errors on the resampled shape is very close to the error on the original shape while recording a significant increase in Dirichlet energy.

7. Conclusion, Limitations, and Future Work

In this work, we introduced a transformer-based pipeline for point-to-point shape correspondence that operates without large-scale labeled datasets and directly interprets cross-attention weights as correspondence estimates. Through a careful architectural design and a sampling strategy that incorporates both vertices and interior points, our method captures the continuous nature of surfaces and achieves robust performance across a diverse set of shape pairs.

Limitations. Despite its strengths, our approach has some limitations. Most notably, the method relies on a small set of semantic landmarks, which currently play a crucial role in guiding the correspondence estimation, we show however that our model can work with other semantically meaningful features such as those of Diff3f. As a result, in challenging shape pairs, especially those exhibiting strong non-isometries or significant structural differences, our model may produce inaccurate matches. Additionally, our implementation requires much more computational time than baselines, highlighting the need for more efficient attention computation.

Future Directions. Our work naturally opens several promising avenues for future research. First, it can be integrated into a training framework and used as a direct competitor to the methods of Trappolini et al. [TCM*21] and Raganato et al. [RPM23], with the expectation of improved efficiency due to its lightweight design. Second, an analysis on the dependency on the number and on the noise of landmarks can serve to better test the robustness of the method. Furthermore, an exciting related direction involves pretraining a semantic feature extractor through self-supervised or task-specific learning to eliminate the need for manually provided landmarks or data-driven features such as Diff3f. Third, we recognize that random selection is not the optimal sampling strategy. Future investigations into the selection process are required, for example, to account for local measures and the gradient magnitude at the previous epoch. Ultimately, we aim to investigate the generalization capabilities of our model across various shape representations, as our architecture is already applicable to any representation

where we can sample points (e.g., meshes, point clouds, implicit fields), thereby further enhancing its applicability in wider geometric scenarios.

Acknowledgments

This work was partially supported by the computational resources provided by hpc-ReGAIN@DISCo, which is a MUR Department of Excellence Project within the Department of Informatics, Systems and Communication at the University of Milan-Bicocca (<https://www.disco.unimib.it>), by the PRIN project GEOPRIDE, 2022-NAZ-0115, CUP H53D23003400001, by the NVIDIA Academic Hardware Grant with two RTX A5000 GPUs granted to the University of Milano-Bicocca for the project “Learned representations for implicit binary operations on realworld 2D-3D data”, and by ISCRA awarding this project access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy). Open access publishing facilitated by Università degli Studi di Milano-Bicocca, as part of the Wiley - CRUI-CARE agreement.

References

- [ADH*21] ARNAB A., DEGHANI M., HEIGOLD G., SUN C., LUČIĆ M., SCHMID C.: Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), pp. 6816–6826. 3
- [ASC11] AUBRY M., SCHLICKWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 1626–1633. 3
- [BKL*16] BOGO F., KANAZAWA A., LASSNER C., GEHLER P., ROMERO J., BLACK M. J.: Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European conference on computer vision* (2016), Springer, pp. 561–578. 6
- [BMR*20] BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., AGARWAL S., HERBERT-VOSS A., KRUEGER G., HENIGHAN T., CHILD R., RAMESH A., ZIEGLER D., WU J., WINTER C., HESSE C., CHEN M., SIGLER E., LITWIN M., GRAY S., CHESSE B., CLARK J., BERNER C., MCCANDLISH S., RADFORD A., SUTSKEVER I., AMODEI D.: Language models are few-shot learners. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 1877–1901. 3
- [CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. *Acm transactions on graphics (tog)* 28, 3 (2009), 1–12. 9
- [CRB23] CAO D., ROETZER P., BERNARD F.: Unsupervised learning of robust spectral shape matching. *ACM Trans. Graph.* 42, 4 (jul 2023). 3, 6
- [Dao24] DAO T.: Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations* (2024), vol. 2024, pp. 35549–35562. 11
- [DCO22] DONATI N., CORMAN E., OVSIANIKOV M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. *CVPR* (2022). 3, 10
- [DLR*20] DYKE R. M., LAI Y.-K., ROSIN P. L., ZAPPALÀ S., DYKES S., GUO D., LI K., MARIN R., MELZI S., YANG J.: Shrec’20: Shape correspondence with non-isometric deformations. *Computers & Graphics* 92 (2020), 28–43. 6
- [DMM24] DUTT N. S., MURALIKRISHNAN S., MITRA N. J.: Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 4494–4504. 5, 9

- [Dos20] DOSOVITSKIY A.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020). 3
- [H*23] HUANG Q., ET AL.: Transformers in medical imaging: A survey. *Medical Image Analysis* (2023). 3
- [HDZ*23] HE J., DENG J., ZHANG T., ZHANG Z., ZHANG Y.: Hierarchical shape-consistent transformer for unsupervised point cloud shape correspondence. *IEEE Transactions on Image Processing* 32 (2023), 2734–2748. 4
- [HLC20] HOLZSCHUH B., LÄHNER Z., CREMERS D.: Simulated annealing for 3d shape correspondence. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 252–260. 3
- [HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum* 39, 5 (2020), 265–278. 3
- [HYW*24] HUANG H., YUAN S., WEN C., HAO Y., FANG Y.: 3d-trans: 3d hierarchical transformer for shape correspondence learning. In *2024 10th International Conference on Automation, Robotics and Applications (ICARA)* (2024), IEEE, pp. 536–540. 4
- [J*21a] JI Y., ET AL.: Dnabert: pre-trained bidirectional encoder representations from transformers for dna-language in genome. *Bioinformatics* (2021). 3
- [J*21b] JUMPER J., ET AL.: Highly accurate protein structure prediction with alphafold. *Nature* (2021). 3
- [JGB*21] JAEGLER A., GIMENO F., BROCK A., VINYALS O., ZISSERMAN A., CARREIRA J.: Perceiver: General perception with iterative attention. In *International conference on machine learning* (2021), PMLR, pp. 4651–4664. 3, 4
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 79. 3
- [L*22] LI Y., ET AL.: Competition-level code generation with alphacode. In *Science* (2022). 3
- [LH22] LI Y., HARADA T.: Non-rigid point cloud registration with neural deformation pyramid. *Advances in Neural Information Processing Systems* 35 (2022), 27757–27768. 3, 6
- [LLD*21] LI Z., LIU X., DRENKOW N., DING A., CREIGHTON F. X., TAYLOR R. H., UNBERATH M.: Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (October 2021), pp. 6197–6206. 2
- [LLG*26] LIN J., LONG H., GUO H., ZHANG J., YANG J., GUO T., YANG Y., LI J., ZHANG W., NIESSNER M., YANG W.: Meshripple: Structured autoregressive generation of artist-meshes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2026). 3
- [LLP*25] LIN Y., LIN C., PAN P., YAN H., FENG Y., MU Y., FRAGKIADAKI K.: Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. In *Advances in Neural Information Processing Systems (NeurIPS)* (2025). 3
- [LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* 34, 6 (Oct. 2015), 248:1–248:16. 6
- [LRB*16] LÄHNER Z., RODOLA E., BRONSTEIN M. M., CREMERS D., BURGHARD O., COSMO L., DIECKMANN A., KLEIN R., SAHILLIOĞLU Y., ET AL.: Shrec'16: Matching of deformable shapes with topological noise. In *Eurographics Workshop on 3D Object Retrieval, EG 3DOR* (2016), Eurographics Association, pp. 55–60. 6
- [LZX*24] LIANG D., ZHOU X., XU W., ZHU X., ZOU Z., YE X., TAN X., BAI X.: Pointmamba: A simple state space model for point cloud analysis. In *Advances in Neural Information Processing Systems* (2024). 3
- [MBRM24] MAGGIOLI F., BAIERI D., RODOLÀ E., MELZI S.: Re-matching: Low-resolution representations for scalable shape correspondence. In *European Conference on Computer Vision* (2024), Springer, pp. 183–200. 6, 11
- [MOR*18] MELZI S., OVSJANIKOV M., ROFFO G., CRISTANI M., CASTELLANI U.: Discrete time evolution process descriptor for shape analysis and matching. *ACM Transactions on Graphics (TOG)* 37, 1 (Jan. 2018), 4:1–4:18. 3
- [MRMO20] MARIN R., RAKOTOSAONA M.-J., MELZI S., OVSJANIKOV M.: Correspondence learning via linearly-invariant embedding. NIPS '20, Curran Associates Inc. 3
- [MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)* 38, 6 (Nov. 2019), 155:1–155:14. 3, 6
- [MSP*23] MÜLLER N., SIDDIQUI Y., PORZI L., BULO S. R., KONTSCIEDER P., NIESSNER M.: Diffirf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4328–4338. 2, 3
- [NMR*18] NOGNENG D., MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. *Computer Graphics Forum* 37, 2 (2018), 179–190. 3
- [NO17] NOGNENG D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum* 36, 2 (2017), 259–267. 3
- [OBGS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 30:1–30:11. 3, 6
- [PKO22] PANINE M., KIRGO M., OVSJANIKOV M.: Non-isometric shape matching via functional maps on landmark-adapted bases. In *Computer graphics forum* (2022), vol. 41, Wiley Online Library, pp. 394–417. 3
- [PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: 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)* (Los Alamitos, CA, USA, jun 2021), IEEE Computer Society, pp. 384–393. 3
- [PRW94] PARDALOS P. M., RENDL F., WOLKOWICZ H.: The quadratic assignment problem: A survey and recent developments. 3
- [PWT*22] PANG Y., WANG W., TAY F. E., LIU W., TIAN Y., YUAN L.: Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II* (2022), Springer, pp. 604–621. 3
- [R*19] RADFORD A., ET AL.: Language models are unsupervised multitask learners. *OpenAI Blog* (2019). 3
- [R*20] RONG Y., ET AL.: Self-supervised graph transformer on large-scale molecular data. In *NeurIPS* (2020). 3
- [RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. *Computer Graphics Forum* 40, 5 (2021), 81–96. 3
- [RPM23] RAGANATO A., PASI G., MELZI S.: Attention and positional encoding are (almost) all you need for shape matching. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, p. e14912. 2, 3, 4, 12
- [RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)* 37, 6 (2018). 7
- [RR07] RAHIMI A., RECHT B.: Random features for large-scale kernel machines. *Advances in neural information processing systems* 20 (2007). 5

- [RRM24] RIVA A., RAGANATO A., MELZI S.: Localized Gaussians as Self-Attention Weights for Point Clouds Correspondence. In *Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference* (2024), Caputo A., Garro V., Giachetti A., Castellani U., Dulecha T. G., (Eds.), The Eurographics Association. doi:10.2312/stag.20241345. 2, 4, 5
- [RSCB22] ROETZER P., SWOBODA P., CREMERS D., BERNARD F.: A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 428–438. 3
- [S*19] SCHWALLER P., ET AL.: Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *Chemical Science* (2019). 2
- [SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (ToG)* 41, 3 (2022), 1–16. 2, 5, 9
- [SAL*24] SU J., AHMED M., LU Y., PAN S., BO W., LIU Y.: Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063. 4
- [SKHL24] SHIN J., KIM Y., HONG S., LEE J.: Learning dual hierarchical representation for 3d surface reconstruction. In *Proceedings of the Asian Conference on Computer Vision (ACCV)* (December 2024), pp. 4422–4438. 3
- [SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum* 28, 5 (2009), 1383–1392. 3
- [TCM*21] TRAPPOLINI G., COSMO L., MOSCHELLA L., MARIN R., MELZI S., RODOLÀ E.: Shape registration in the time of transformers. *Advances in Neural Information Processing Systems* 34 (2021), 5731–5744. 2, 3, 4, 5, 12
- [TSDS10] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *Proc. ECCV* (2010), Springer, pp. 356–369. 3
- [VLB*17] VESTNER M., LÄHNER Z., BOYARSKI A., LITANY O., SLOSSBERG R., REMEZ T., RODOLA E., BRONSTEIN A., BRONSTEIN M., KIMMEL R., ET AL.: Efficient deformable shape correspondence via kernel matching. In *2017 international conference on 3D vision (3DV)* (2017), IEEE, pp. 517–526. 3
- [VLP*] VIGANÒ G., LONGARI G., PEREIRA L. F., MIOLANE N., MELZI S.: Geomfum: A python package for machine learning with functional maps. 6
- [VLR*17] VESTNER M., LITMAN R., RODOLÀ E., BRONSTEIN A., CREMERS D.: Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proc. CVPR* (2017), pp. 6681–6690. 3
- [VOM25] VIGANÒ G., OVSJANIKOV M., MELZI S.: Nam: Neural ad-joint maps for refining shape correspondences. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–15. 3, 6
- [VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. *Advances in neural information processing systems* 30 (2017). 2, 3
- [WJW*24] WU X., JIANG L., WANG P.-S., LIU Z., LIU X., QIAO Y., OUYANG W., HE T., ZHAO H.: Point transformer v3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024). 3
- [WLJ*22] WU X., LAO Y., JIANG L., LIU X., ZHAO H.: Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems* (2022). 3
- [YRW*26] YUE Y., ROBERT D., WANG J., HONG S., WEGNER J. D., RUPPRECHT C., SCHINDLER K.: LitePT: Lighter Yet Stronger Point Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2026). 3
- [ZGJM21] ZHANG Z., GIRDHAR R., JOULIN A., MISRA I.: Self-supervised pretraining of 3d features on any point-cloud. In *ICCV* (2021). 3
- [ZJJ*21] ZHAO H., JIANG L., JIA J., TORR P. H., KOLTUN V.: Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 16259–16268. 3
- [ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D. W., BLACK M. J.: 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 6365–6373. 6