Neural Semantic Surface Maps

Luca Morreale[1] and Noam Aigerman[2,3] and Vladimir G. Kim[3] and Niloy J. Mitra[1,3]

[1]Unversity College London
[2]University of Montreal
[3]Adobe Research

arXiv:2309.04836v3 [cs.CV] 8 Mar 2024

Figure 1: *Here, we show the extracted map between two non-isometric shapes,* `tiger` *and* `iguana`*. Although the shapes are highly non-isometric, e.g., lengths of the tail and legs, our method successfully associated these regions between shapes, thus yielding a semantically correct map. This map is seamless by construction, and optimized with no supervision thanks to pre-trained ViT models which can identify semantically corresponding points across shape renderings.*

Abstract
We present an automated technique for computing a map between two genus-zero shapes, which matches semantically corresponding regions to one another. Lack of annotated data prohibits direct inference of 3D semantic priors; instead, current state-of-the-art methods predominantly optimize geometric properties or require varying amounts of manual annotation. To overcome the lack of annotated training data, we distill semantic matches from pre-trained vision models: our method renders the pair of untextured 3D shapes from multiple viewpoints; the resulting renders are then fed into an off-the-shelf image-matching strategy that leverages a pre-trained visual model to produce feature points. This yields semantic correspondences, which are projected back to the 3D shapes, producing a raw matching that is inaccurate and inconsistent across different viewpoints. These correspondences are refined and distilled into an inter-surface map by a dedicated optimization scheme, which promotes bijectivity and continuity of the output map. We illustrate that our approach can generate semantic surface-to-surface maps, eliminating manual annotations or any 3D training data requirement. Furthermore, it proves effective in scenarios with high semantic complexity, where objects are non-isometrically related, as well as in situations where they are nearly isometric.

CCS Concepts
• Computing methodologies → Shape analysis; Mesh geometry models; Feature selection;

1. Introduction

In this work, we propose an automatic method to compute a continuous correspondence between two genus-zero surfaces, represented as meshes. Our core contribution is an approach for computing a *semantic* map that matches semantically corresponding points to one another (e.g., nose to nose, arm to arm, etc.).

Computing correspondences between domains is a fundamental and highly-researched problem, spanning a wide array of domains such as text snippets [HDT19], audio [ZLW*20], images [MJF*21], or general graphs [SZF20]. In the context of 3D surfaces, establishing such correspondences enables texture or deformation transfer [SP04, BVGP09], shape analy-

sis [TP91, BRPM*16, BRLB14, PWH*15], and shape space exploration [HWAG09, MCA*22, YYPM11].

Continuous surfaces (2-manifolds), encoded as triangular meshes, remain the most natural and common representation of 3D shapes in graphics and discrete differential geometry. Correspondence between two such surfaces is typically required to be a map that is continuous, one-to-one, onto, and with a continuous inverse, *i.e.*, a *homeomorphism*. Decades of research (see surveys [Sah20, EST*19]) have been dedicated to tackle the task of mapping between surface pairs. These previous works, being geometric, cannot extract (semantic) maps over the space of homeomorphisms; instead, they have focused on surrogate optimization tasks that minimize some geometric notion of "distortion" of the map, *e.g.*, preserving geodesic distances as best as possible. Such distortion-minimizing geometrically-guided maps are, of course, not necessarily semantically meaningful. Thus, a human-in-the-loop approach is usually taken to manually indicate landmark correspondences, which are then used to optimize a map.

Computing semantic homeomorphism faces two main challenges. First, the lack of annotated 3D data inhibits learning high-level semantic priors. In contrast, considering the image domain, recent works [SPV*21, HZH*22, VHVZ22, AGBD21, ODM*23] demonstrate that the features of a pre-trained vision transformer (ViT) are often semantically meaningful and can be used reliably across multiple vision tasks, even on out-of-training image data in a zero-shot setting. Second, most 3D representations either hinder or - completely - prevent the computation of bijective inter-surface maps from semantic priors. We aim to bridge the semantic matching capabilities of the image domain with the computation of inter-surface maps from potentially noisy correspondences, encouraging continuity and bijectivity. Our core observation is that suitable renderings of the surfaces, without access to surface texture, are already sufficient for image transformers (i.e., ViT) to produce 2D matches that can subsequently be used as fuzzy (i.e., partial and non-injective) maps between the surfaces. Then, we formulate an optimization to aggregate multiple such fuzzy matches obtained from multiview renderings to produce a surface map that best conforms to these fuzzy matches, thereby distilling their semantic priors, see Figure 1.

Specifically, given the fuzzy matches, we utilize Neural Surface Maps (NSM) [MAKM21] to optimize a map between two surfaces. The original NSM framework encodes surfaces using dedicated neural functions, offering a differentiable backbone and avoiding complexities arising from topological changes (i.e., mesh connectivity for different triangulations). However, it has two limitations: it expects the individual surfaces to be cut into disc topologies with the two respective boundaries *already* in correspondence and requires a set of exact landmark correspondences. We address the first problem by proposing a seamless Neural Surface Maps (sNSM) framework, which relaxes the requirement from exact boundary correspondences to only cone-point matchings. We address the second problem by optimizing a custom objective that encourages the image of a specific point to best accommodate the fuzzy (semantic) matches while identifying and disregarding outliers (see Figure 2). The resultant optimization problem is solved using gradient descent, simply through PyTorch's SGD optimizer.
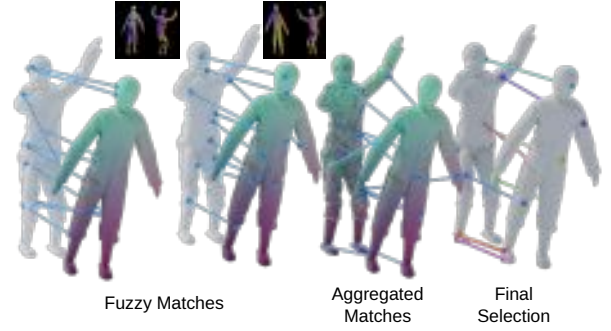

Fuzzy Matches   Aggregated Matches   Final Selection

Figure 2: *Fuzzy semantic correspondences. (Left) We lift 2D image-based correspondences, obtained using a pre-trained vision-transformer [ODM*23] on rendered image source/target pairs from sampled views, to obtain fuzzy and spurious 3D (semantic) correspondences. We collect correspondence, shown with coloring and a random set highlighted with lines, from each of the sampled views and aggregate them across views to get aggregated fuzzy matches (middle), which contain erroneous matching, e.g., thigh getting mapped to the arm. (Right) We propose an optimization to distill these fuzzy matches into an inter-surface map, here depicting a subset of matches closer than a given threshold ($d < 0.1$) wrt the optimized map.*

Through quantitative and qualitative experiments, we evaluate our ability to match upright object pairs with varying levels of isometry for objects from the same semantic class and across different ones. We also compare ours to competing surface map extraction algorithms. In summary, our main contributions are:

- proposing a fully automatic algorithm for extracting semantic maps between upright shape pairs;
- sampling and integrating a set of image-based correspondences to form fuzzy object space correspondence maps;
- extending Neural Surface Maps framework to seamless maps that can work with fuzzy, and potentially noisy guidance, to distill semantic maps; and
- demonstrating, via extensive evaluation and comparisons, that the algorithm yields semantically valid maps for both isometrically and non-isometrically related shape pairs.

2. Related Works

2.1. Shape Matching

Shape matching and correspondence estimation have been widely studied in geometry processing. In the simplest setting where rigid transforms relate shape pairs, iterative closest points methods [RL01, ZSN03] are often used for local refinement while relying on deep learned features to identify good correspondences to provide an initial global alignment [GZWW19, DBI18, WS19]. Directly optimizing for distance preservation is computationally challenging for surfaces [BBK06, HAWG08] discretized as dense triangulations. Optimal transport has been used to optimize over relatively coarse shape representations [SPKS16].

Functional maps compute a fuzzy correspondence by aligning the spectral basis of two shapes that are related by linear transforms for near-isometric deformations [OBCS*12, CSBC*17]. The

approach offers elegant machinery to compute shape correspondence and has been extended to handle partial matching [LRB*18], learn spectral alignment [LRR*17, DSO20], find multiple low-distortion maps [RMOW20], and leverage optimal transport methods [PRM*21] (c.f. course notes [OCB*17] for other variations). While variants have been proposed to 'project' functional maps to point-to-point maps, such approaches [EBC17] do not explicitly ensure bijectivity of the maps.

Restricting optimizations directly to surfaces poses a challenge since the most common surface representation, a triangle mesh, does not easily lend itself to continuous optimization. Schreiner *et al.* [SAPH04] optimize surface-to-surface maps via direct mesh-mesh intersection, which leads to a combinatorial problem due to the surface discretization into triangles. Subsequent approaches represent maps by parameterizing meshes to a common domain such as a plane [APL14, SBCK19] or a sphere [APH05, SPK23], which can give a bijective final map, but still does not offer a natural way to optimize using inter-surface distortion. Notably, Schmidt *et al.* [SPK23] propose a coarse-to-fine optimization strategy, defined between spheres, that uses incremental re-meshing to significantly speed up the optimization and improve the quality of the map, yet relies on few manual annotations.

Others treated inter-surface mapping as a problem of exploring a space of low-distortion maps, for example, blending across conformal maps, Blended Intrinsic Maps (BIM) [KLF11] were explored and interpolated. The recently proposed Enigma [EEBC20] leverages genetic algorithms to achieve state-of-the-art results. In another notable effort, the MapTree [RMOW20], authors explore multiple functional maps between near-isometric shape pairs. They propose a fully automatic method that reveals multiple and diverse maps by first enumerating map variations and then optimizing them to extract dense pointwise maps.

Parallel to this work, Abdelreheem *et al.* [AEOW23] proposed a pipeline to extract coarse correspondences between shapes through pre-trained networks. In particular, the authors use Blip2 [LLSH23] to extract a shape class description, *e.g.*, human or person; such class is then used to extract semantic descriptions for shape sub-parts, *e.g.*, leg or arm. Finally, SAM [KMR*23] extracts features based on these keywords and renderings to extract features for each shape. The resulting features are then used for co-segmentation and shape correspondences. The latter is achieved through the functional map framework [RPWO18], which natively produces a fuzzy map, and thus may not sufficiently reduce, or refine, the fuzziness of the initial set of correspondences.

Instead of functional maps, which assume that a linear spectral map exists and is less suitable for optimizing bijectivity, we build on NSM [MAKM21] that maps shapes through common 2D domains. However, the original technique implicitly assumes shapes to have corresponding boundaries, making it unsuitable for fully automatic mapping. Thus, we modify the formulation to be seamless across any cut boundary, allowing for arbitrary non-matching cuts on different surfaces. We rely only on DinoV2 [ODM*23] to extract fuzzy features from shape renderings to produce point-level (rather than part-level) correspondence priors.

2.2. Image-based Shape Analysis

Image-based representations are commonly used for 3D shape analysis tasks, such as classification [SMKL15], segmentation [SYM*22, KYF*20, DLH22], or matching [HKC*17], where a shape is first rendered from multiple viewpoints, the resulting images are analyzed with 2D neural networks, and then the output is aggregated on the 3D shape via additional optimization [KAMC17]. While these methods often start with pre-trained 2D neural networks, they require additional fine-tuning with 3D supervision and thus can only work on categories of shapes with labeled 3D data. Indeed, Genova et al. [GYK*21], train a 3D segmentation technique by using a 2D method to produce pseudo labels. In a concurrent effort, [ASOW23] describes a training-free approach for 3D shape semantic segmentation using pre-trained visual transformers. Our approach exploits a similar intuition to extract semantic shape correspondences, distilling an inter-surface map from them, thus *without* any 3D supervision.

2.3. Visual Features

The use of pre-trained CNNs features marked a vital milestone for computer vision tasks, such as object detection and segmentation [GDDM14], or image synthesis [GEB16, SGM*20]. These network representations encode a wide range of visual information from low-level (statistical) features, (*e.g.*, edges, auto-correlation matrices), object parts, and structure [OMS17, CAS*19, MGY*19]. However, these methods [LLUZ16] usually are restricted to local (CNN) neighborhoods or pre-authored nonlocal receptive field, and ignore long-range dependencies [WGGH18].

Vision Transformers [DBK*20], dubbed ViT, belong to a family of recent and powerful neural architecture that can discover both local and nonlocal relations. A noteworthy example is DINO-ViT [CTM*21] that trains a transformer network through self-distillation and uses its features in several tasks, *e.g.*, image retrieval and object segmentation. Several works demonstrated the utility of Dino-ViT internal representation as a black box [SPV*21, WSH*22] for tasks such as semantic segmentation [HZH*22] and category discovery [VHVZ22]. Amir *et al.* [AGBD21] study these features and use them to solve vision tasks, such as image correspondences, in zero-shot settings. Recently, Oquab *et al.* [ODM*23] extended Dino-ViT, introducing Dinov2, showcasing enhanced feature semantic interpretability compared to the original version, and also exhibiting broader applicability.

3. Method

We now detail our framework (see Figure 3) for an automatic inter-surface map. We assume to be given two upright 3D surfaces, **A** and **B**, in arbitrary relative poses. The majority of meshes from online repositories, such as the 3D Warehouse, TurboSquid, or Sketchfab, satisfy this requirement, alternatively, methods like [PLDZ22, PJQ*20] can be used as pre-processing. We assume both shapes to have zero genus, although the method can be extended to higher genus surfaces. We aim to compute an inter-surface map $\Psi : \mathbf{A} \leftrightarrow \mathbf{B}$ guided by visual semantics. Our framework proceeds in three stages:
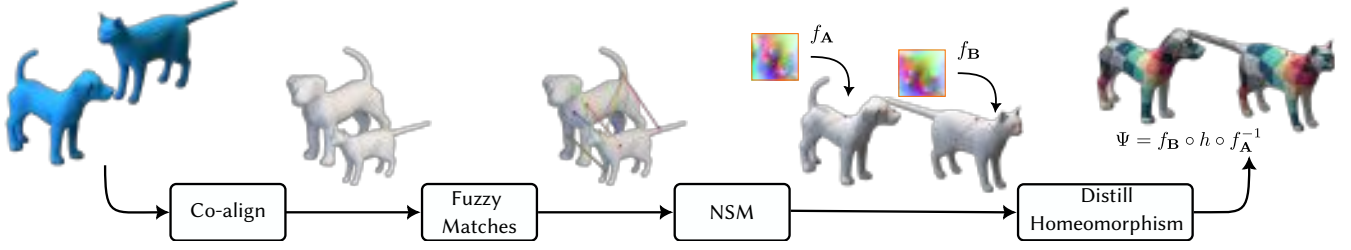
Figure 3: ***Overview.*** *Starting from a pair of upright genus-zero surfaces, we automatically distill an inter-surface map from a set of fuzzy matches. First, we align the input shapes, then extract a set of fuzzy matches through DinoV2 [ODM*23] semantic visual features. We use these features to independently cut the two meshes and then optimize a (seamless) map between them.*

(i) Given two shapes, **A** and **B**, that are assumed to be oriented up-right, we automatically align them using semantic matches.

(ii) We aggregate fuzzy matches (i.e., general matching of pairs of points which is neither 1-to-1, onto, nor maps all points on the source surface) between the surfaces by applying 2D matching techniques to renderings, over multiple views.

(iii) Optimize a surface map that best agrees with the fuzzy semantic matches while handling outliers.

3.1. Semantic Shape Alignment

Given two upright shapes, **A** and **B**, we first align them to have the same orientations. We achieve this by casting this problem as (semantic) circular string matching between shape renderings: given two 'strings' – sets of renderings – of the same length, we find the global rotation **R**, about the upaxis, to best align one string with the other. Intuitively, we order one sequence to convey semantic information in the same order as the other, see Figure 4 for an overview.

First, we render each mesh from 12 viewpoints around it, $R_i^{\mathbf{A}}$ and $R_i^{\mathbf{B}}$ (see Subsection 3.4 for a discussion on rendering). These images constitute the two strings $s^{\mathbf{A}} := \{R_i^{\mathbf{A}}\}$ and $s^{\mathbf{B}} := \{R_i^{\mathbf{B}}\}$. Then, we extract a set of DinoV2 [ODM*23] features for each image. Finally, we compute the alignment score for the 12 possible rotations as the total number of "Best Buddy matches" [DOR*15] between the two strings of features. We pick the (relative) rotation with the highest score as the rotation and use it to co-align the shapes.

3.2. Distilling Fuzzy 3D Correspondences via Visual Semantics

Next, we extract fuzzy matches from renderings, taken from different viewpoints, of the aligned surfaces. Each such viewpoint V results in a pair of rendering that we use to define a fuzzy match $\phi^V := \left(p_i^V, q_i^V\right)_{i=1}^{n}$ with $p \in \mathbf{A}, q \in \mathbf{B}$, which consists of pairs of corresponding points on **A** and **B**.

Although matches are imprecise or inaccurate, we assume that these imprecisions balance out, leading to approximately correct matches. Embracing this assumption, we leverage it as a guiding principle during map optimization.

Computing rendering matches. Given a viewpoint V, we render the two untextured surfaces from that viewpoint to get two renderings, $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$. To extract matches, we take inspiration from
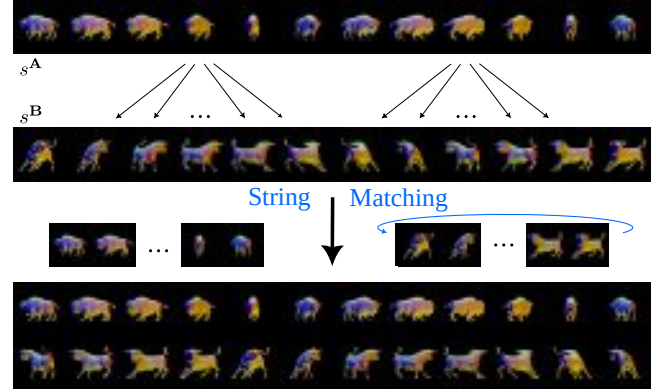

Figure 4: ***Co-aligning input surfaces:*** *Starting from a pair of upright meshes (`bison` and `bull` in this example), we render 12 views around them ($s^{\mathbf{A}}$ and $s^{\mathbf{B}}$). Then, we extract DinoV2 features from each rendering independently and match these features as a string-matching problem. Specifically, we optimize over a cyclic shift of the rendered views (i.e., one degree of freedom) to maximize agreement of image-based semantic correspondences.*

recent methods that leverage deep image features from [ODM*23] for matching 2D images and design a method for extracting dense visual matches. Specifically for each image patch processed by Dinov2, we extract a feature vector with $\lambda_i^{\mathbf{A}}$ and $\lambda_i^{\mathbf{B}}$ being the features of rendering of $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$, respectively. Then, we segment foreground/background through PCA and compute the cosine similarity between all pairs of source and target patch foreground features, as score

$$S_{ij} = \left\langle \lambda_i^{\mathbf{A}}, \lambda_j^{\mathbf{B}} \right\rangle. \tag{1}$$

Finally, we define the match of patch $i \in R_V^{\mathbf{A}}$ as the patch $j \in R_V^{\mathbf{B}}$ with the highest cosine similarity, and vice versa, the match of patch $j \in R_V^{\mathbf{B}}$ as the patch $i \in R_V^{\mathbf{A}}$ with the highest cosine similarity. In summary, the pair $(i, j), i \in R_V^{\mathbf{A}}, j \in R_V^{\mathbf{B}}$ is a match, if

$$S_{ij} = \max_k S_{ik} \text{ or } S_{ij} = \max_l S_{lj}. \tag{2}$$

We transform a match from patch level to pixel level, as the patch size is known. In contrast to ours, [AGBD21] selects only "Best Buddy" matches [DOR*15], augments features with binning, and does not segment foreground/background through PCA features.

Although [AGBD21] produces a more expressive set of features and possibly a more reliable set of fuzzy matches, we found it time-consuming (2hrs in our settings), and our experiments did not provide sufficient justification for such a design choice.

Given dense 2D matches in an image, we lift (unproject) each pixel to the 3D mesh by performing ray intersection between that pixel's corresponding ray from viewpoint V and the 3D mesh $\mathbf{T}$, thereby associating every 2D pixel with a point on the surface, represented as barycentric coordinates at the triangle the ray intersects. The fuzzy correspondences are thus pairs of matching 3D points (represented as barycentric coordinates on triangles):
$$\phi^V := \left(p_i^V, q_i^V \right)_{i=1}^{n}.$$

We repeat this process from multiple viewpoints and obtain a collection $\left\{ \phi^i \right\}_{i=1}^{k}$ of fuzzy correspondences. Our final task is to distill them to produce an automatic map.

3.3. Aggregating the Fuzzy Correspondences to an Inter-surface map

Given the fuzzy correspondences, we wish to optimize a continuous map Ψ between $\mathbf{A}$ and $\mathbf{B}$ using a differentiable loss that encourages agreement with the fuzzy correspondences.

Our final goal is thus to devise an optimization scheme that will lead to a map $\Psi : \mathbf{A} \leftrightarrow \mathbf{B}$ which balances smoothness with the number of respected correspondences. To achieve this goal, we compare each point's image with its designated corresponding point from ϕ^i with the L1 norm. Then, for a set of corresponding pairs (p_j, q_j), we minimize the average error as follows:
$$\mathcal{L}_{\text{Matches}} = \frac{1}{N} \sum_{j=1}^{N} \|\Psi(p_j) - q_j\|_1, \qquad (3)$$

where N is the number of correspondences. By averaging these distances, we encourage sparsity of correspondences. To optimize Eq. 3, we adopt a recent method for optimization of the surface map, Neural Surface Maps (NSM) [MAKM21] as described next.

Seamless Neural Surface Map. We follow NSM's paradigm: we first parameterize each one of the two cut surfaces via SLIM [RPPSH17] into a square $D \in \mathbb{R}^2$ to get two bijective seamless parameterizations, $P_\mathbf{A} : \mathbf{A} \leftrightarrow D, P_\mathbf{B} : \mathbf{B} \leftrightarrow D$. Then, we fit a neural network to each of the two parameterizations' inverse, $f_\mathbf{A} \approx P_\mathbf{A}^{-1}, f_\mathbf{B} \approx P_\mathbf{B}^{-1}$. Finally, using another neural network that maps the square to itself, h, we can define the inter-surface map $\Psi = f_\mathbf{B} \circ h \circ f_\mathbf{A}^{-1}$. By optimizing solely the parameters of h while maintaining its bijectivity, and holding the overfitted networks $f_\mathbf{A}, f_\mathbf{B}$ fixed, NSM enables optimization over the space of maps between the two surfaces.

As we cannot guarantee corresponding cuts between genus 0 meshes, see cut examples in Figure 5, we relax the boundary-matching constraint in the original NSM and extend it to support seamless maps. Intuitively, a borderless, or seamless, parameterization is a 2D-3D mapping that is independent of the choice of cut path, given a set of K boundary points. In other words, the map emerging from the parametrization has several equivalent maps with different boundaries, see Figure 6(c). Only the K

points, referred to as cones, remain constant and must have the same mapping across all equivalent maps. Mathematically, a seamless parametrization is a mapping equipped with homotopic cuts (i.e., the cuts can be changed homotopically but the produced mapping will stay the same). In particular, for three cones on a sphere, all cuts are homotopic, and thus the embedding is independent of the cut choice. Please refer to [APL15] for more details.

Furthermore, the class of seamless parametrization requires a specific type of cut such that triangles, or points for that matter, can be mapped to the other side of the cut by a family of transformations $\mathcal{R}$. In terms of NSM, a seamless map requires matching corresponding cones while the boundary is allowed to move. Thus, the accuracy required to define the cut path, and hence the 2D boundary, through fuzzy matches is reduced, *e.g.*, see Figure 5(c) for cut paths. Below, we first detail how we extract corresponding cones, then describe a seamless map.

Cones. To identify cones, we first aggregate the fuzzy correspondences by counting for each triangle $F_i^\mathbf{A} \in \mathbf{A}$, how many fuzzy correspondences associate it with triangle $F_j^\mathbf{B} \in \mathbf{B}$, yielding a large sparse matrix M such that its (i, j) entry is the total count for correspondences of $F_i^\mathbf{A}$ to $F_j^\mathbf{B}$,
$$M_{ij} = \sum_k \left| \left\{ (p, q) \text{ s.t. } p \in F_i^\mathbf{A}, q \in F_j^\mathbf{B}, (p, q) \in \phi^k \right\} \right|, \qquad (4)$$

where $|\cdot|$ stands for the cardinality of the set. Next, we consider M as the adjacency matrix of an edge-weighted graph, with M_{ij} being the weight on edge (i, j). Then, through bipartite graph matching [HK73] we obtain a matching, *i.e.*, a list of pairs (i_k, j_k), s.t. $i, j = \underset{i,j}{\mathrm{argmax}}\, M_{i,j}$. We select the $K = 3$ correspondences (i, j) with the highest M_{ij} values, such that the geodesic distance - averaged between the two shapes - between all K points is at least $\tau = 0.3$. Finally, we use these landmarks as the cut's endpoints and the midpoint.

Seamlessness. Since we cannot rely on the cut quality, we reformulate the neural map h, which we optimize to define the map Ψ, to support seamlessness. This constrains the map to work on shape pairs with the same genus. Furthermore, the definition of the seam-
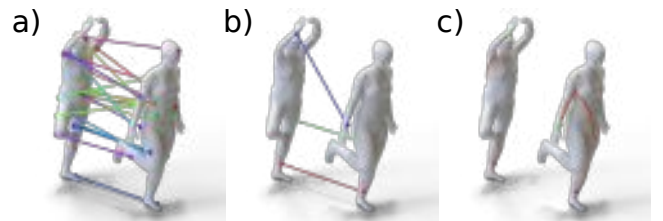

Figure 5: *Cutting through cone points. We collect a set of spurious and noisy matches (a). Then, we select the most reliable $K = 3$ correspondences (b). Finally, using these correspondences as cut endpoints, or cones, we cut the two meshes independently (c). Note how the cut differs in the two shapes: the man is cut through the back, while the woman is cut through the front. Refer to Sec. 3.3 for details.*
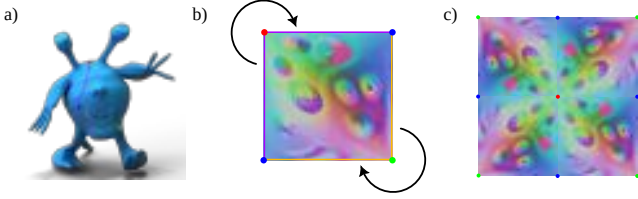
a)
b)
c)

Figure 6: *Seamless cuts. To parametrize a genus-zero mesh (a) we cut and map it to a disc topology, cut visualized as in (b). The two corresponding sides of the cut match perfectly, i.e., when we connect the two parts, the map remains continuous across the cut (c).*

less map changes based on the genus. For a sphere, h changes to $\tilde{h}$:

$$\tilde{h} = \left\{ x \to T \cdot h(x) + \eta \,\middle|\, T = \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \in \mathbb{R}^{2 \times 2}, \eta \in \mathbb{R}^{2 \times 1} \right\} \quad (5)$$

for all points mapped outside the domain D, which rotates around the cone c_i (η) of a rotation R (T).

To achieve seamlessness h must perfectly match cones c_i to their ground truth $\tilde{c}_i$. Therefore, we formulate such a constraint by penalizing the deviation of mapped cones, $h(c_i)$, to their ground truth position:

$$\mathcal{L}_{\text{Cones}} = \|h(c_i) - \tilde{c}_i\|. \quad (6)$$

In the case of spheres, we have 3 cones, of which one is duplicated, thus one for each vertex of the square in the square domain D. In the case of torus, a single point is duplicated 4 times, corresponding to all 4 square vertices.

A second condition for seamlessness concerns the duplicated points on the boundary. In the case of spheres, each point on the boundary p_1 has a corresponding point p_2 which is a rotation of $90°$ with respect to one of the cones. For the case of a sphere, we formulate the constraint as the following energy:

$$\mathcal{L}_{\text{Seamless}} = \|h(p_1) - R \cdot (h(p_2) - c_i) + c_i\|, \quad (7)$$

where c_i is the cones wrt p_2 undergoes a rotation R to be a clone of p_1. Note, the rotation can either be $\pi/2$ or $-\pi/2$. In the case of a torus, p_2 is on the opposite side of the boundary of p_1, i.e., the transformation being a translation along x or y.

Optimization energies. We follow NSM and encourage the map to be bijective through a loss term that prevents the map h's Jacobian J_p at every point $p \in D$ from having a negative determinant:

$$\mathcal{L}_{\text{J}} = \int_D \max\left(-sign\left(|J_p|\right) e^{-|J_p|}, 0\right). \quad (8)$$

Thus, encouraging, but not guaranteeing, continuity and bijectivity of the map.

To cope with the sparsity of fuzzy matches and obtain a well-defined map in undefined regions, we use an energy term that encourages smoothness and prevents large distortion:

$$\mathcal{L}_{\text{Smooth}} = \int_D \left\| J_p^{\Psi} - J_{p^{\varepsilon}}^{\Psi} \right\|, \quad (9)$$

where J_p^{Ψ} is the Jacobian at a point p of the map Ψ. While p^{ε} is

the point p perturbed by $\varepsilon \sim \mathcal{N}(0, 0.1)$ through barycentric coordinates. Intuitively, we want the Jacobian of the map to change slowly. Note, in NSM [MAKM21], the authors used Symmetric Dirichlet [RPPSH17] in a similar context, however, this energy promotes isometric maps rather than smooth ones. Such behavior can actively damage the map optimization and force it to ignore certain - correct - matches, while we aim to attend to unregularized areas.

Total energy. Our total loss is expressed as:

$$\mathcal{L} = \alpha_1 \mathcal{L}_{\text{matches}} + \alpha_2 \mathcal{L}_{\text{J}} + \alpha_3 \mathcal{L}_{\text{Cones}} + \\ \alpha_4 \mathcal{L}_{\text{Seamless}} + \alpha_5 \mathcal{L}_{\text{Smooth}}, \quad (10)$$

where $\alpha_1 = 10^4$, $\alpha_2 = 10^6$, $\alpha_3 = 10^6$, $\alpha_4 = 10^6$, and $\alpha_5 = 10^{-1}$ in all experiments. These hyper-parameters were selected experimentally. We optimize network weights h using this loss, and to alleviate the impact of incorrect matches; we incrementally drop those that strongly disagree with the current map, *i.e.*, 20% of matches with the highest Euclidean distance. Experimentally, this explicit filtering reduces the impact of incorrect matches, thus preventing the network from getting stuck in incorrect energy minima due to the presence of inaccurate matches.

In summary, the proposed pipeline optimizes for an inter-surface map *automatically* between a pair of upright shapes. First, the input mesh pair is automatically aligned, then through pre-trained ViT [ODM*23] we extract a large set of semantic fuzzy matches between them. Finally, we distill an inter-surface map; this step is fundamental to filter out any incorrect matches, enhancing the overall accuracy and reliability of the resultant map. Next, we quantitatively and qualitatively evaluate the quality of the distilled map.

3.4. Rendering Settings

We render shape pairs and use these images with Dino-ViT2 [ODM*23]; this model is known to be forgiving in cases of image variation. As the shape alignment is unknown, we render an object-centric scene with a fixed perspective camera and 5 point lights aimed at the shape. Different points of view are obtained by rotating *only* the shape by fixed increments, while the rest of the scene (*i.e.*, camera and lights) stays fixed. We set up the scene to ensure the entire object is visible by the camera's field of view.

To boost the matching capabilities of Dino-ViT and aid it in distinguishing left from right, top from bottom, while enhancing scene details, we strategically position colored lights around the object in a half-dome fashion. Specifically, we employ five colored point lights (red, blue, green, yellow, and white) for this purpose. As depicted in Figure 4, corresponding regions in the images exhibit similar colors; for instance, the right part of the images tends to appear reddish due to illumination from a red light source. In cases involving textured meshes, we replace the colored lights with white ones.

3.5. Implementation details

Following NSM [MAKM21], we never require to compute Ψ to optimize $\mathcal{L}$. To evaluate $\mathcal{L}_{\text{Matches}}$ we first compute the barycentric coordinates of p_j and convert it to a point in the square, $p_j^{2D} \in \mathbb{R}^2$. Then, this point is mapped forward through $f^{\mathbf{B}} \circ h$ and used to

Figure 7: *Results. Automatic maps extracted by the optimization on various surface pairs using aggregated fuzzy correspondences. Colored landmarks and paths show automatically selected cones and cuts by our method. The rabbit, hands, humans, and heads examples represent near isometric pairs with pose variations; the chairs, giraffe-horse, giraffe-cow examples produce non-isometric mappings with spatially varying distortions. Note the semantic nature of the extracted maps. No explicit energy term was used to encourage the maps to be isometric.*

compute the error as $\|f^{\mathbf{B}}(h(p_j^{2D})) - q_j\|_1$ for each correspondence. Furthermore, as the number of correspondences is extremely large $N \approx 65k$, at runtime we estimate $\mathcal{L}_{\text{Matches}}$ on a subset ($M << N$). We follow a similar strategy for $\mathcal{L}_{Seamless}$ by precomputing a set of p_j^{2D} from the boundary and pushing them forward $f_B \circ h$. Differently, $\mathcal{L}_{Smooth}$ and $\mathcal{L}_J$ require only the computation of Jacobians which can be estimated from forward maps. Finally, $\mathcal{L}_{Cones}$ penalizes the prediction of known 2D points, thus requiring only h. During the evaluation, we rely on h using $P_B \circ h \circ P_A^{-1}$, see supplementary material for more detail.

In all experiments, we define the neural map (h) as a 4-layer residual MLP of 128 neurons each, while neural surfaces (f) are always 8 layers residual MLP with 256 neurons. While training, we sample 1024 points to enforce injectivity and smoothness, and 128 points on the boundary to enforce seamlessness and $M = 128$ correspondences in each iteration.

4. Evaluation

We evaluated our method on various datasets for inter-surface mapping and compared it against multiple baselines that focus on obtaining surface-to-surface maps.

Table 1: *Quantitative evaluation. We compute each map's accuracy (i.e., average geodesic error) and averaged them over 30 shape pairs for each dataset.*

	FAUST			SHREC07			SHREC19		
	Inv ↓	Bij ↓	Acc ↓	Inv ↓	Bij ↓	Acc ↓	Inv ↓	Bij ↓	Acc ↓
ICP	0.06	0.17	0.25	0.09	0.65	**0.23**	0.07	0.75	0.15
BIM	0.09	0.03	0.04	0.49	0.48	**0.23**	0.05	0.82	0.04
Zoomout	0.33	0.23	0.15	0.25	0.65	0.54	0.29	0.76	0.32
Smooth-shells	0.01	**0.00**	**0.01**	0.03	0.72	0.26	0.01	0.83	**0.01**
Ours	**0.00**	**0.00**	0.13	**0.00**	**0.00**	0.23	**0.00**	**0.00**	0.11

Datasets We assess maps' quality on available benchmarks comprising isometric and non-isometric shape pairs. (i) We randomly select 30 pairs from FAUST [BRLB14], containing isometric deformations and pose variations of human shapes. (ii) We choose 30 random same-category shape pairs from SHREC07 [GBP07], containing non-isometric deformations across multiple categories of shapes. (iii) We also extract 30 random shape pairs among the listed test set of SHREC19 from Dyke *et al.* [DSLR19], containing a mix of isometric and non-isometric deformations.

To ablate the effect of pose variation, we use FAUST [BRLB14], SCAPE [ASP*04], and TOSCA [BBK08]. To ablate the effects of rendering settings and rotation, we use FAUST [BRLB14]; 3DBiCar [LCD*23], which comprise a variety of textured shapes; and SHREC15 [LZEE*15], which contain significant non-isometric-variations, with manually-annotated sparse correspondences. In the supplementary material, we present additional ablations highlighting the crucial role of initial alignment, the method's robustness to mesh holes and nois, and discuss Dinov2 features. To summarize, significant misalignment negatively impacts matching quality; feature similarity does not reflect their matching accuracy; finally, the method effectively maps meshes with holes, *e.g.*, scans.

All meshes used in our experiment are watertight and genus zero, and range from 11K to 90K faces. The shape pairs include a mix of some isometric and mostly non-isometric cases.

Metrics We assess map quality (see also [RPWO18]) based on their accuracy, bijectivity, and inversion as:

- *Accuracy (Acc ↓):* measures the ability of the algorithms to respect ground-truth correspondences. We measure it as the geodesic distance normalized, as defined in [KLF11], for each landmark.
- *Bijectivity (Bij ↓):* measures the geodesic distance of all vertices mapped forward, and then back to the source mesh wrt their original position. A zero value indicates perfect bijectivity.

Figure 8: *Comparisons. Left-to-right: Source model, results using BIM [KLF11], ICP, Smooth-shells [ELC20], ZoomOut [MRR*19], and Ours. Although geometric methods produce good maps, they often yield discontinuous maps, e.g., see the wings of planes. Ours explicitly encourages continuity and bijectivity. Colored landmarks and paths show* automatically *selected cones and cuts by our method. Note that our maps are continuous across the cut seams. No explicit energy term is used to encourage isometric maps.*
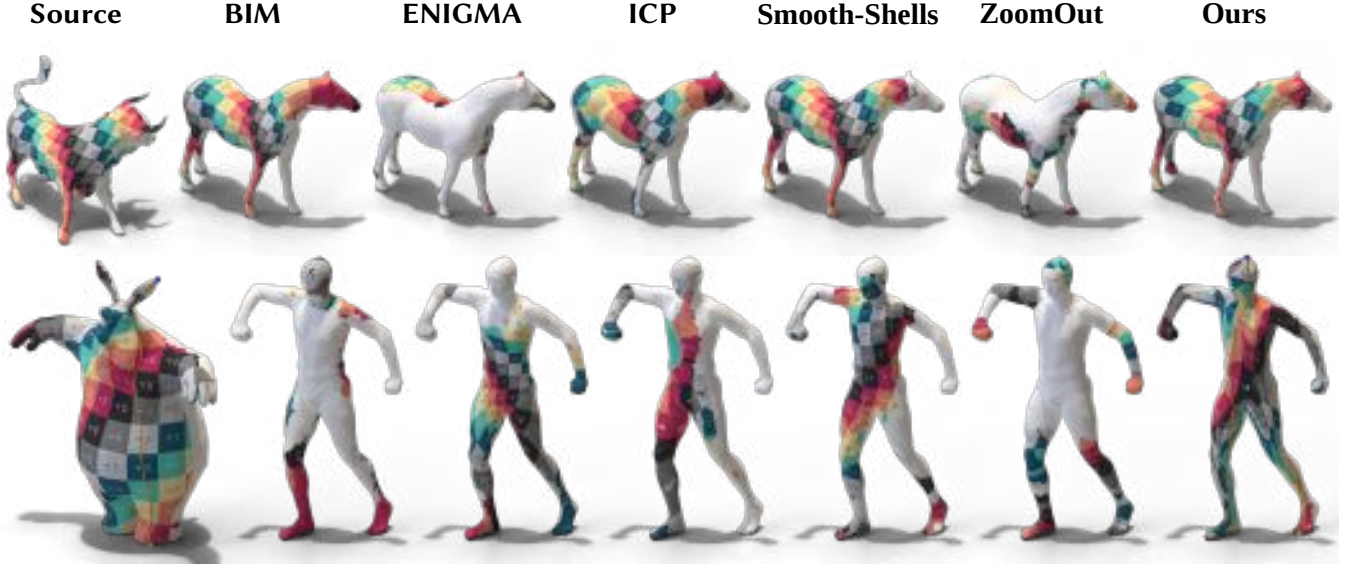
Figure 9: *Qualitative comparison. ENIGMA [EEBC20] fails to produce correct mappings, in cases of extreme deformations. Similarly, other state-of-the-art methods may lack bijectivity or correct correspondence. Ours can better handle these cases, see Table 1 for quantitative comparison. Colored landmarks and paths show* automatically *selected cones and cuts by our method.*

- *Inversion (Inv ↓):* measures how often the map flips the surface as the percentage of inverted triangles; we compute it as the agreement of the normal of the mapped triangles wrt the faces on which the triangle vertices are mapped.

Baselines We compare with three other techniques that focus on extracting maps between given surfaces: (i) BIM [KLF11], (ii) Zoomout [MRR*19], and (iii) Smooth-shells [ELC20]. We also include (iv) ICP, which uses the closest points as correspondence, as a strawman approach that performs well in case of negligible pose variation. Results are presented in Table 1, and selection of the pairs shown in Figure 8.

We cast ICP as nearest neighbor search after rigid alignment. Specifically, we use our pipeline first to align each shape pair and then compute nearest neighbor correspondences for each point on the source to the target mesh. This approach may perform well for shapes in similar poses with low isometric deformations.

Additionally, we compare qualitatively to Enigma [EEBC20] that uses genetic algorithms along with a combinatorial search to find a set of good sparse correspondence, which are then interpolated to a dense low-distortion map. While this method produces smoother and more semantic maps than other baselines, it still suffers from large and uneven distortions, see Figure 9.

4.1. Qualitative Evaluation

Figure 7 shows Neural Semantic maps extracted using our fully automatic approach. The produced maps accurately match semantic features despite the fuzzy aggregated correspondences being erroneous and confused by symmetries (*e.g.*, mapping incorrect limbs). Ours also work well across dissimilar shapes. These non-isometric cases require introducing significant local stretching to

preserve semantic correspondence. The extracted maps still exhibit low isometric distortion, where possible, while adhering to semantically meaningful correspondences. Yet, artifacts may arise (see Armadillo's leg in Figure 7) when the smoothing energy is not sufficient to balance the noisiness of matches. State-of-the-art methods, such as ENIGMA [EEBC20] or Smooth-shells [ELC20], suffer from self-symmetry ambiguities, *e.g.*, see `bull-horse` in Figure 9.

Aggregation To assess the importance of the map distillation module, we present a qualitative comparison in Figure 10 with the method proposed by Surface Maps via Adaptive Triangulations (SMAT) [SPK23], where we replace manual correspondences with automatically extracted ones. As the original approach requires a set of bijective correspondences, we randomly subsample a set of $N = 64$ matches from the automatically extracted ones to ensure consistency, *i.e.*, no vertex appears twice. Then, we optimize for a bijective map that respects these landmarks. We refer to it as Dinov2+SMAT. Note SMAT [SPK23] optimize for isometric energy (Dirichlet), while we optimize only for smoothness, see Eq 9.

As SMAT does not account for inaccurate nor imprecise correspondences, it is unable to filter out wrong correspondences. In our observations, optimizing a map with the original hyperparameters leads to visible inversions. This issue arises from SMAT's attempt to preserve all landmarks, resulting in maps with extreme stretches, a phenomenon intensified by the discrete nature of meshes. Adaptive remeshing struggles to handle these extreme stretches effectively, leading to visibly distorted maps. To mitigate this effect, we trade landmark precision for map continuity and quality. As shown in Figure 10, although both maps appear continuous, [SPK23] is unable to filter out inaccurate correspondences and yield a reasonable map. Note that this experiment mainly assesses the importance
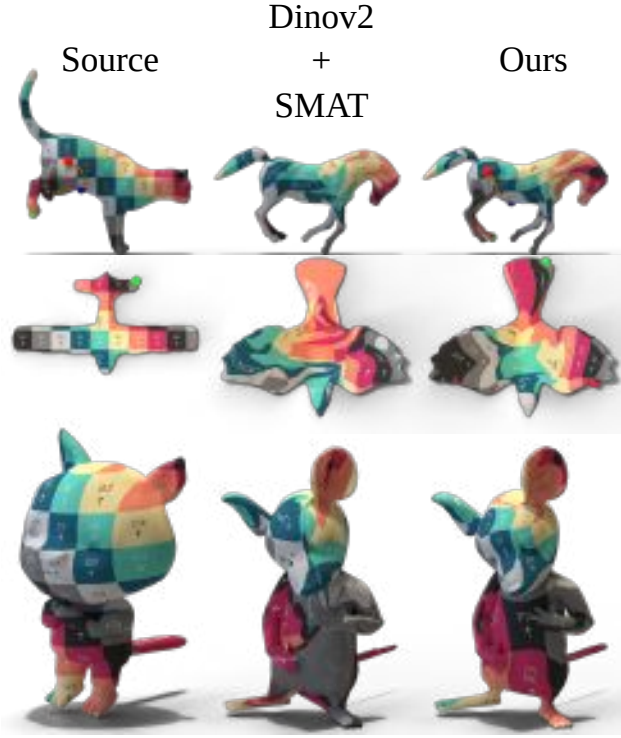
Figure 10: *Qualitative comparison. Surface Maps via Adaptive Triangulations (SMAT) [SPK23] optimize for bijective and continuous maps, relying on manual annotations. We pair it with Dinov2, Dinov2+SMAT, by replacing these manual annotations with $k = 64$ automatically extracted ones, $\{\phi^i\}$ with $i = 1 : k$, then we optimize the inter-surface map to construct an automatic inter-surface map. While Dinov2+SMAT attempts to satisfy all correspondences together with bijectivty, ours automatically filters out incorrect correspondences, yielding a more continuous and semantically correct map.*

of our correspondence distillation step, and Dinov2+SMAT is not the mode SMAT was originally designed for. Ours, without any explicit isometric or conformal energy term, still can produce smooth and semantics-respecting bijective maps.

4.2. Quantitative Evaluation

We report quantitative errors using the metrics discussed earlier. In particular, for accuracy, we follow the standard practice and measure the mean geodesic distance to ground truth correspondence on a unit-area mesh.

Although not guaranteed by construction, we empirically found that ours consistently offers more bijective and continuous maps, see Table 1 "*Bij*" and "*Inv*", while others can fail to perfectly achieve these properties in both isometric and non-isometric cases. Our technique shows comparable quality in the maps in non-isometric cases (SHREC07) compared to state-of-the-art methods, Table 1 "*Acc*", while it performs worse in isometric cases (FAUST and SHREC19). In general, our method suffers in these cases as it

does not exploit geometric cues and does not have an explicit isometric energy term, thus producing less accurate maps than competing methods.

5. Limitations

Timing. A key limitation of our method is its long running time. The map optimization takes on average 1.5 hours, converting the meshes into their neural representation which requires about 1 hour, and extracting all Dino-ViT matches takes 21 minutes. We plan to investigate approaches such as Meta-Learning and better caching to speed up this process.

Occlusion. The presence of self-occlusion in shape pairs prevents DinoViT from correctly mapping regions across shapes, thus consistently making mistakes. We believe incorporating other priors, or an advanced rendering pipeline (e.g., layered rendering) may help cope with this issue.

Thin parts. We struggle to handle thin parts, as our pipeline requires parameterizing objects. Specifically, thin parts are difficult to handle unless cut points are manually placed.

6. Conclusion

We have presented a method that produces a semantic surface-to-surface map guided by visual semantic priors, by computing it from a set of candidate non-injective and discontinuous partial maps extracted by matchings over renderings of untextured 3D surfaces. Our method has many potential practical applications, ranging from matching scans of human faces and bodies to clothes, anatomical scans, and archaeological findings. These depend on the quality of the matchings achieved over the renderings of objects from these categories, which we aim to explore.

Future work. We require surfaces to be cut as required by NSM [MAKM21] which makes our method more prone to error. We aim to improve the existing pipeline to avoid cutting altogether by replacing the 2D disks with 3D spheres [GGS03], as successfully used in [SPK23]. Our optimization cannot guarantee achieving a global optimum nor that the global optimum defines the "most-meaningful" semantic map, and we mark extending our method to directly *learn* to produce maps from a dataset as an important future direction. Our method can create such a dataset, augmented with manual input to score the goodness of any extracted semantic map. We believe this work is only a step in producing semantic-driven maps. Candidate fuzzy maps extracted from other means can be considered. For instance, methods to predict fuzzy geometric correspondences directly over 3D surfaces trained for specific tasks can alternatively produce fuzzy maps and can be used in conjunction with semantic and/or visual cues.

References

[AEOW23] ABDELREHEEM A., ELDESOKEY A., OVSJANIKOV M., WONKA P.: Zero-shot 3d shape correspondence. *arXiv preprint arXiv:2306.03253* (2023). 3

[AGBD21] AMIR S., GANDELSMAN Y., BAGON S., DEKEL T.: Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814* (2021). 2, 3, 4, 5, 13, 15, 16

[APH05] ASIRVATHAM A., PRAUN E., HOPPE H.: Consistent spherical parameterization. In *Computer Graphics and Geometric Modeling (CGGM) 2005 Workshop* (2005). 3

[APL14] AIGERMAN N., PORANNE R., LIPMAN Y.: Lifted bijections for low distortion surface mappings. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 1–12. 3

[APL15] AIGERMAN N., PORANNE R., LIPMAN Y.: Seamless surface mappings. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–13. 5

[ASOW23] ABDELREHEEM A., SKOROKHODOV I., OVSJANIKOV M., WONKA P.: Satr: Zero-shot semantic segmentation of 3d shapes. *arXiv preprint arXiv:2304.04909* (2023). 3

[ASP*04] ANGUELOV D., SRINIVASAN P., PANG H.-C., KOLLER D., THRUN S., DAVIS J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. *Advances in neural information processing systems 17* (2004). 7

[BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing* (2006). 2

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 7

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 3794–3801. 2, 7, 15

[BRPM*16] BLACK M., ROMERO J., PONS-MOLL G., MAHMOOD N., BOGO F.: Learning human body shapes in motion. In *ACM SIGGRAPH 2016 Courses* (2016), SIGGRAPH '16. 2

[BVGP09] BARAN I., VLASIC D., GRINSPUN E., POPOVIĆ J.: Semantic deformation transfer. *ACM Trans. Graph. 28*, 3 (2009). 1

[CAS*19] CARTER S., ARMSTRONG Z., SCHUBERT L., JOHNSON I., OLAH C.: Activation atlas. *Distill 4*, 3 (2019), e15. 3

[CCC*08] CIGNONI P., CALLIERI M., CORSINI M., DELLEPIANE M., GANOVELLI F., RANZUGLIA G.: MeshLab: an Open-Source Mesh Processing Tool. In *Eurographics Italian Chapter Conference* (2008), Scarano V., Chiara R. D., Erra U., (Eds.), The Eurographics Association. 16

[CSBC*17] CORMAN E., SOLOMON J., BEN-CHEN M., GUIBAS L., OVSJANIKOV M.: Functional characterization of intrinsic and extrinsic geometry. *ACM Trans. Graph. 36*, 2 (mar 2017). URL: https://doi.org/10.1145/2999535, doi:10.1145/2999535. 2

[CTM*21] CARON M., TOUVRON H., MISRA I., JÉGOU H., MAIRAL J., BOJANOWSKI P., JOULIN A.: Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 9650–9660. 3, 14, 15, 16

[DBI18] DENG H., BIRDAL T., ILIC S.: Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 195–205. 2

[DBK*20] DOSOVITSKIY A., BEYER L., KOLESNIKOV A., WEISSENBORN D., ZHAI X., UNTERTHINER T., DEHGHANI M., MINDERER M., HEIGOLD G., GELLY S., ET AL.: An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020). 3

[DLH22] DECATUR D., LANG I., HANOCKA R.: 3d highlighter: Localizing regions on 3d shapes via text descriptions. *arXiv preprint arXiv:2212.11263* (2022). 3

[DOR*15] DEKEL T., ORON S., RUBINSTEIN M., AVIDAN S., FREEMAN W. T.: Best-buddies similarity for robust template matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 2021–2029. 4

[DSLR19] DYKE R., STRIDE C., LAI Y., ROSIN P.: Shrec-19: shape correspondence with isometric and non-isometric deformations. 7

[DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601. 3

[EBC17] EZUZ D., BEN-CHEN M.: Deblurring and denoising of maps between shapes. In *Symposium on Geometry Processing* (2017). 3

[EEBC20] EDELSTEIN M., EZUZ D., BEN-CHEN M.: Enigma: Evolutionary non-isometric geometry matching. In *ACM Transactions on Graphics* (2020). 3, 9

[ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274. 8, 9, 14

[EST*19] EGGER B., SMITH W. A. P., TEWARI A., WUHRER S., ZOLLHÖFER M., BEELER T., BERNARD F., BOLKART T., KORTYLEWSKI A., ROMDHANI C., THEOBALT C., BLANZ V., VETTER T.: 3d morphable face models - past, present and future. *CoRR abs/1909.01815* (2019). URL: http://arxiv.org/abs/1909.01815. 2

[GBP07] GIORGI D., BIASOTTI S., PARABOSCHI L.: Shrec: shape retrieval contest: Watertight models track. *Online]: http://watertight. ge. imati. cnr. it 7* (2007). 7

[GDDM14] GIRSHICK R., DONAHUE J., DARRELL T., MALIK J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 580–587. 3

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2414–2423. 3

[GGS03] GOTSMAN C., GU X., SHEFFER A.: Fundamentals of spherical parameterization for 3d meshes. *ACM Trans. Graph. 22*, 3 (jul 2003), 358–363. 10

[GYK*21] GENOVA K., YIN X., KUNDU A., PANTOFARU C., COLE F., SUD A., BREWINGTON B., SHUCKER B., FUNKHOUSER T.: Learning 3d semantic segmentation with only 2d image supervision. *3DV* (2021). 3

[GZWW19] GOJCIC Z., ZHOU C., WEGNER J. D., WIESER A.: The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 5545–5554. 2

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-Rigid Registration Under Isometric Deformations. *Computer Graphics Forum* (2008). doi:10.1111/j.1467-8659.2008.01285.x. 2

[HDT19] HU W., DANG A., TAN Y.: A survey of state-of-the-art short text matching algorithms. In *International Conference on Data Mining and Big Data* (2019). 1

[HK73] HOPCROFT J. E., KARP R. M.: An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing 2*, 4 (1973), 225–231. 5

[HKC*17] HUANG H., KALOGERAKIS E., CHAUDHURI S., CEYLAN D., KIM V. G., YUMER E.: Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics 37*, 1 (2017). 3

[HWAG09] HUANG Q., WICKE M., ADAMS B., GUIBAS L.: Shape decomposition using modal analysis. In *Computer Graphics Forum (Proceedings of Eurographics 2009)* (Munchen, Germany, April 2009), vol. 28, pp. 407–416. doi:10.1111/j.1467-8659.2009.01380.x. 2

[HZH*22] HAMILTON M., ZHANG Z., HARIHARAN B., SNAVELY N., FREEMAN W. T.: Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414* (2022). 2, 3

[JSR*22] JAKOB W., SPEIERER S., ROUSSEL N., NIMIER-DAVID M., VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org. 13

[KAMC17] KALOGERAKIS E., AVERKIOU M., MAJI S., CHAUDHURI S.: 3D shape segmentation with projective convolutional networks. In *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)* (2017). 3

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *Transactions on Graphics (Proc. of SIGGRAPH) 30*, 4 (2011). 3, 7, 8, 9, 14, 15

[KMR*23] KIRILLOV A., MINTUN E., RAVI N., MAO H., ROLLAND C., GUSTAFSON L., XIAO T., WHITEHEAD S., BERG A. C., LO W.-Y., ET AL.: Segment anything. *arXiv preprint arXiv:2304.02643* (2023). 3

[KYF*20] KUNDU A., YIN X., FATHI A., ROSS D., BREWINGTON B., FUNKHOUSER T., PANTOFARU C.: Virtual multi-view fusion for 3d semantic segmentation. *ECCV* (2020). 3

[LCD*23] LUO Z., CAI S., DONG J., MING R., QIU L., ZHAN X., HAN X.: Rabit: Parametric modeling of 3d biped cartoon characters with a topological-consistent dataset. *arXiv preprint arXiv:2303.12564* (2023). 7, 15

[LLSH23] LI J., LI D., SAVARESE S., HOI S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597* (2023). 3

[LLUZ16] LUO W., LI Y., URTASUN R., ZEMEL R.: Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems 29* (2016). 3

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG) 34*, 6 (2015), 1–16. 16

[LRB*18] LITANY O., RODOLA E., BRONSTEIN A., BRONSTEIN M., CREMERS D.: Partial single-and multishape dense correspondence using functional maps. *Handbook of Numerical Analysis* (2018). 3

[LRR*17] LITANY O., REMEZ T., RODOLA E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 5659–5667. 3

[LZEE*15] LIAN Z., ZHANG Z., EL ELNAGHY H., EL SANA J., FURUYA T., GIACHETTI A., GÜLER A., LAI L., LI C., LI H., ET AL.: Shrec 15 track non rigid 3d shape retrieval. 7, 15

[MAKM21] MORREALE L., AIGERMAN N., KIM V. G., MITRA N. J.: Neural surface maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4639–4648. 2, 3, 5, 6, 10, 14

[MCA*22] MURALIKRISHNAN S., CHAUDHURI S., AIGERMAN N., KIM V., FISHER M., MITRA N.: Glass: Geometric latent augmentation for shape spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022). 2

[MGY*19] MO K., GUERRERO P., YI L., SU H., WONKA P., MITRA N., GUIBAS L.: Structurenet: Hierarchical graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575* (2019). 3

[MJF*21] MA J., JIANG X., FAN A., JIANG J., YAN J.: Image matching from handcrafted to deep features: A survey. *Int. J. Comput. Vision 129*, 1 (2021), 23–79. 1

[MRR*19] MELZI S., REN J., RODOLA E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865* (2019). 8, 9, 14

[OBCS*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), 1–11. 2

[OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLA E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. *SIGGRAPH Course Notes* (2017). 3

[ODM*23] OQUAB M., DARCET T., MOUTAKANNI T., VO H., SZAFRANIEC M., KHALIDOV V., FERNANDEZ P., HAZIZA D., MASSA F., EL-NOUBY A., ET AL.: Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023). 2, 3, 4, 6, 13, 14, 15, 16

[OMS17] OLAH C., MORDVINTSEV A., SCHUBERT L.: Feature visualization. *Distill 2*, 11 (2017), e7. 3

[PJQ*20] POURSAEED O., JIANG T., QIAO H., XU N., KIM V. G.: Self-supervised learning of point clouds via orientation estimation. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 1018–1028. 3

[PLDZ22] PANG X., LI F., DING N., ZHONG X.: Upright-net: Learning upright orientation for 3d point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 14911–14919. 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 384–393. 3

[PWH*15] PISHCHULIN L., WUHRER S., HELTEN T., THEOBALT C., SCHIELE B.: Building statistical shape spaces for 3d human modeling. *CoRR abs/1503.05860* (2015). URL: http://arxiv.org/abs/1503.05860. 2

[RL01] RUSINKIEWICZ S., LEVOY M.: Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling* (2001), IEEE, pp. 145–152. 2

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: Maptree: recovering multiple solutions in the space of maps. *ACM Trans. Graph. 39*, 6 (2020), 264–1. 3

[RPPSH17] RABINOVICH M., PORANNE R., PANOZZO D., SORKINE-HORNUNG O.: Scalable locally injective mappings. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1. 5, 6

[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), 1–16. 3, 7

[Sah20] SAHILLIOĞLU Y.: Recent advances in shape correspondence. *The Visual Computer 36*, 8 (2020), 1705–1721. 2

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. In *ACM SIGGRAPH 2004 Papers*. 2004, pp. 870–877. 3

[SBCK19] SCHMIDT P., BORN J., CAMPEN M., KOBBELT L.: Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–15. 3

[SGM*20] SHOCHER A., GANDELSMAN Y., MOSSERI I., YAROM M., IRANI M., FREEMAN W. T., DEKEL T.: Semantic pyramid for image generation. In *Proc. CVPR* (2020), pp. 7457–7466. 3

[SMKL15] SU H., MAJI S., KALOGERAKIS E., LEARNED-MILLER E. G.: Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV* (2015). 3

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (aug 2004), 399–405. 1

[SPK23] SCHMIDT P., PIEPER D., KOBBELT L.: Surface Maps via Adaptive Triangulations. *Computer Graphics Forum* (2023). doi:10.1111/cgf.14747. 3, 9, 10

[SPKS16] SOLOMON J., PEYRÉ G., KIM V. G., SRA S.: Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (ToG) 35*, 4 (2016), 1–13. 2

[SPV*21] SIMÉONI O., PUY G., VO H. V., ROBURIN S., GIDARIS S., BURSUC A., PÉREZ P., MARLET R., PONCE J.: Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279* (2021). 2, 3

[SYM*22] SHARMA G., YIN K., MAJI S., KALOGERAKIS E., LITANY O., FIDLER S.: Mvdecor: Multi-view dense correspondence learning for fine-grained 3d segmentation. In *ECCV* (2022). 3

[SZF20] SUN H., ZHOU W., FEI M.: A survey on graph matching in computer vision. In *Intn. Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)* (2020), pp. 225–230. 1

[TP91] TURK M., PENTLAND A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience 3*, 1 (01 1991), 71–86. 2

[VHVZ22] VAZE S., HAN K., VEDALDI A., ZISSERMAN A.: Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 7492–7501. 2, 3

[WGGH18] WANG X., GIRSHICK R., GUPTA A., HE K.: Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 7794–7803. 3

[WS19] WANG Y., SOLOMON J. M.: Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 3523–3532. 2

[WSH*22] WANG Y., SHEN X., HU S. X., YUAN Y., CROWLEY J. L., VAUFREYDAZ D.: Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 14543–14553. 3

[YYPM11] YANG Y.-L., YANG Y.-J., POTTMANN H., MITRA N. J.: Shape space exploration of constrained meshes. *ACM Transactions on Graphics 30*, 6 (2011). 2

[ZLW*20] ZHU H., LUO M., WANG R., ZHENG A., HE R.: Deep audio-visual learning: A survey, 2020. URL: https://arxiv.org/abs/2001.04758, doi:10.48550/ARXIV.2001.04758. 1

[ZSN03] ZINSSER T., SCHMIDT J., NIEMANN H.: A refined icp algorithm for robust 3-d correspondence estimation. In *Proceedings 2003 international conference on image processing (Cat. No. 03CH37429)* (2003), vol. 2, IEEE, pp. II–695. 2

Appendix A: Pseudocode

We provide pseudocode for our semantic homeomorphic map extraction framework in Algorithm 1.

Algorithm 1: Semantic Surface Homeomorphism

Data: source $\mathbf{A}$, target $\mathbf{B}$
$\mathbf{R} \leftarrow$ COALIGN(DinoViT(), $\mathbf{A}, \mathbf{B}$) ;
fuzzyMatches $\leftarrow$
  COMPUTEMATCHES(DinoViT(), $\mathbf{A}, \mathbf{B}, \mathbf{R}$) ;
$A_{disk}, B_{disk} \leftarrow$ ASYNCCUT($\mathbf{A}, \mathbf{B}$, fuzzyMatches) ;
$A_{NSM} \leftarrow$ OVERFITNSM(A_{disk}) ;
$B_{NSM} \leftarrow$ OVERFITNSM(B_{disk}) ;
map $\leftarrow$ DISTILMAP(A_{NSM}, B_{NSM}, fuzzyMatches);
return map

Appendix B: Rendering Details

In all cases, we render images of the same size, i.e., 1344×1344 with Mitsuba [JSR*22] using $spp = 150$ and a path integrator. When extracting semantic matches, we limit to rotations around the up-axis (y) - 20 steps between $[0, 2\pi)$ - and forward-axis (z) - 10 steps between $[\frac{-\pi}{2}, \frac{\pi}{2})$ - obtaining 200 images for each shape. Similarly, to align shapes, we rotate around the up-axis - 12 steps - with fixed increments. To uplift 2D pixels to 3D for the matches, we use ray-triangle intersection. On average, we get 328 correspondences per view, totaling 65k correspondences across the 200 views.

Appendix C: Computing rendering correspondences

As discussed in the main manuscript, we render the two surfaces from a given viewpoint to get two renderings, $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$. We leverage DinoV2 [ODM*23] to extract semantic features in the image space, thus obtaining $\lambda_i^{\mathbf{A}}$ and $\lambda_i^{\mathbf{B}}$ as features of rendering of $R_V^{\mathbf{A}}$ and $R_V^{\mathbf{B}}$, respectively. Then, to segment foreground/background we rely on PCA's first component of these features as it naturally groups them in opposite half-spaces.

Finally, we match features with the cosine similarity between all feature pairs from the same viewpoint, as score S_{ij}. We define the match of patch $i \in R_V^{\mathbf{A}}$ as the patch $j \in R_V^{\mathbf{B}}$ with the highest cosine similarity, and vice versa, the match of patch $j \in R_V^{\mathbf{B}}$ as the patch $i \in R_V^{\mathbf{A}}$ with the highest cosine similarity. In summary, the pair $(i, j), i \in R_V^{\mathbf{A}}, j \in R_V^{\mathbf{B}}$ is a match, if

$$S_{ij} = \max_k S_{ik} \text{ or } S_{ij} = \max_l S_{lj}. \tag{11}$$

Patch generation, feature extraction, and PCA

Images are split into (non-overlapping) patches of 14 pixels. Then, Dinov2 [ODM*23] embeds these patches in a forward pass. Following [AGBD21], we use *keys* as feature vectors.

To segment foreground/background we rely on PCA's first component of the features. As discussed in [ODM*23], the features' sign naturally groups them in opposite half-spaces. As the sign is appointed randomly, we use the attention mask from the last layer

to select the correct half-space: we average the first PCA component of the features and take the half-space which agrees with the positive attention mask. Matches are estimated only between foreground patches.

Finally, to unproject a match to 3D, we first translate a patch to a pixel using the known patch size, and then identify the 3D point on each shape (ray casting).
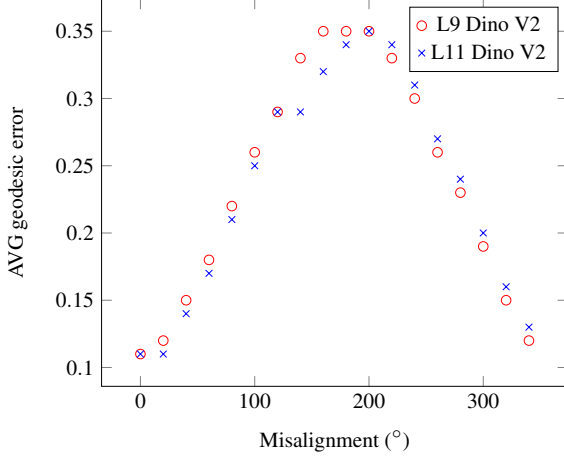

Figure 11: *Robustness to misalignment: the quality of matches depends on the quality of alignment. In the case of severe misalignment ($60°$ or more), we observe poor correspondence.*

Appendix D: Comparison Details

We discuss the main considerations for/against the competing algorithms we compare against.

Blended Intrinsic Maps (BIM) [KLF11] is a classic method that uses geometric priors without any learning component. Namely, it picks a subset of self-consistent and low-distortion conformal maps and then blends them using weighted averages. Individual conformal maps can handle very non-isometric surfaces, however, they can produce high isometric distortion even in near-isometric cases. Note also that the resulting blended map is not a homeomorphism nor even continuous.

Zoomout [MRR*19] and Smooth-shells [ELC20] are both functional maps-based methods. Zoomout starts with a small functional correspondence matrix and iterative upsamples it in the spectral

Table 2: *Dino ViT pose ablation: DinoV2 [ODM*23] matches are significantly more accurate than DinoV1 [CTM*21] in case of pose variation, with no significant difference between features from L9 and L11.*

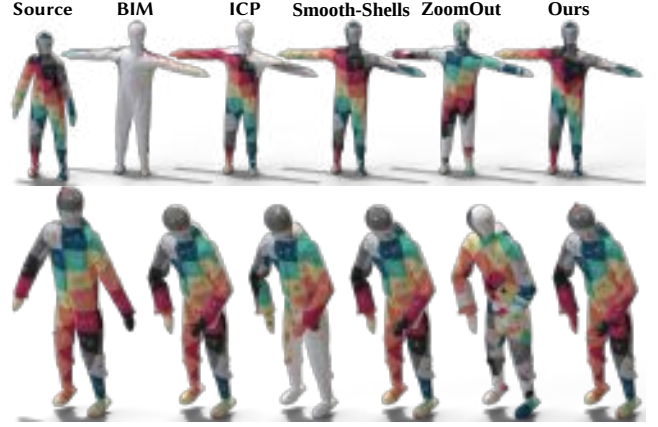| Layer	FAUST		SCAPE		TOSCA	
	9	11	9	11	9	11
DinoV1	0.16	0.16	0.38	0.40	0.27	0.29
DinoV2	0.09	0.09	0.18	0.18	0.27	0.25

Figure 12: *Qualitative comparison SHREC19: Functional maps-based methods produce good maps, although often being discontinuous. Ours explicitly encourages continuity and bijectivity.*

domain. Smooth-shells follow a similar coarse-to-fine scheme, relying on shells as a proxy for functional basis. To handle self-symmetries, Eisenberger et al. [ELC20] incorporate MCMC to evaluate multiple possible functional maps.

We initialize Zoomout's map (C_{21}) as an identity of size 4 as by official implementation. Then, we refine it until it contains 50 eigenvectors. Similarly, for Smooth-shells we follow the official implementation and use MCMC to bootstrap the map using $K_{min} = 6$ and $K_{max} = 20$. and evaluating $N_{prop} = 500$ proposal. In both cases, no landmarks are used. Finally, for ICP we first align the two input shapes as described in Sec. 3.1, and then estimate the nearest neighbor for each vertex.

We depict maps for the different methods on SHREC19 in Figure 12. State-of-the-art methods work well as they exploit geometric cues, although they are susceptible to self-symmetries (see BIM [KLF11] first row). Conversely, "Ours" relies purely on visual cues, with no isometric regularization, thus being less accurate on average.

Appendix E: Differences with Neural Surface Maps

Neural Surface Maps [MAKM21] defines the general mapping framework used to optimize maps. Following the original work, the input two shapes must be homomorphic to a disk with their boundary in correspondence. As this constraint is impossible to satisfy automatically, this work relies on seamless maps, thus relaxing this constraint to 3 corresponding points which are extracted automatically. Furthermore, we define a soft correspondence term to handle inaccurate correspondences, while NSM enforces exact correspondences with an L2 loss over all correspondences.

Appendix F: Metrics

In all experiments, all shapes are automatically normalized and centered.

Table 3: *Dino ViT ablation: DinoV2 [ODM*23] works better than its predecessor [CTM*21], with no significant difference between features from L9 and L11. The use of colored lights (rows DinoV1 and DinoV2) offers better visual cues to extract matches than white lights. Although counter-intuitive, the use of* simple *texture reduces the visual cues available to Dino ViT.*

	FAUST		SHREC15		3DBiCar	
Layer	9	11	9	11	9	11
DinoV1	0.10	0.12	0.32	0.32	0.36	0.49
DinoV2	0.11	0.11	0.24	0.24	0.33	0.33
white lights (V1)	0.20	0.18	0.27	0.35	0.38	0.38
white lights (V2)	0.11	0.11	0.24	0.24	0.30	0.31
texture (V1)	-	-	-	-	0.26	0.26
texture (V2)	-	-	-	-	0.29	0.29

Bijectivity We estimate the map's bijectivity of the shape vertices for all baselines. For ICP, BIM, Zoomout, and Smooth-shells we map all vertices forward ($A \to B$) and then backward ($A \leftarrow B$), using the forward and backward map respectively. Then, we compute the geodesic distance between the starting vertex and its forward-backward map.

Similarly, for consistency we evaluate "Ours" bijectivity only for the shape vertices. In particular, we map a vertex in A onto B's 2D domain through h, and then, we use the piecewise linear map for 2D-3D. For B to A, we pullback vertices through barycentric coordinates after mapping forward all A's triangles. Empirically, for "Ours" *we never observe flips*; while for baselines, correspondences are always given, thus, no ambiguity arises. In the case of a non-bijective map, we would consider the first triangle.

Appendix G: Ablation

On Dinov2 features

As aforementioned, we deem a match if the cosine similarity S_{ij} between patch features - $\lambda_i^\mathbf{A}$ and $\lambda_i^\mathbf{B}$ - is the highest. While this is a common similarity measure, it is important to acknowledge its inherent limitations. Specifically, one notable challenge is that similarity scores derived from different images may not be directly comparable. For example, for two correspondences with scores 0.9 and 0.8, the former match pair is not necessarily better than the latter. In essence, features extracted from one view may be extremely dissimilar to those extracted from another view, even for the same shape. This arises from the inherent variation in image structure across different views and how features are generated from them. This inherent variability hinders consistency in cross-image feature comparisons. Consequently, the process of aggregating features across different views can potentially yield unexpected outcomes, leading to either incorrect matching or highly inaccurate results.

Experimentally, sampling the top $k = 100$ correspondences based on the similarity produced far worse results than uniform sampling or uniform weighting, see Figure 14 for qualitative comparison. In both cases, we optimize maps following the proposed algorithm: *Ours* uses all correspondences, while *TopK* is limited

to $k = 100$ correspondences with the highest similarity score. Visibly, some of these correspondences are incorrect and bias the map towards incorrect minima, thus their similarity score is not representative of their quality. Indeed, the use of all correspondences prevents the map from falling into such a degenerate solution, as the majority of correspondences are reasonably correct.

Tuning DinoVit Matches

We ablate the quality of matches based on DinoViT's degrees of freedom - layer features - in different contexts: pose variation, presence of texture, lights, and misalignment. We conduct our analysis on three distinct datasets: **FAUST** [BRLB14], **3DBiCar** [LCD*23], and **SHREC15** [LZEE*15] each with **dense** or **sparse** ground truth.

We select 12 shape pairs, 4 for each dataset, to ablate texture and misalign concerning the choice of Dino ViT feature layer, as discussed in [AGBD21]. Similarly, we assess the effect of pose variation for the same model with a single instance of FAUST, SCAPE, and TOSCA mapped onto all the other provided poses. We report the quantitative results in Table 3 and show shape pairs examples and qualitative optimization results in Figure 13.

We assess the quality of the aggregated correspondences in terms of the normalized average geodesic distance [KLF11]. We follow the procedure described in the main paper to aggregate the fuzzy correspondences, thus, obtaining a face-wise map M from
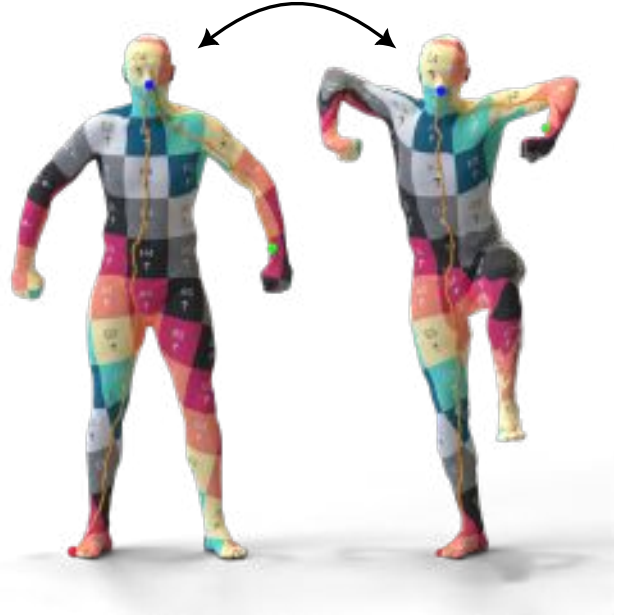

Figure 13: *Pose variation: we assess the ability of DinoV2 [ODM*23] to establish matches between shapes in different poses, as those in the figure. Experimentally, DinoV2 yields correspondences able to guide our pipeline to a proper solution. Colored landmarks and paths show* automatically *selected cones and cuts by our method.*
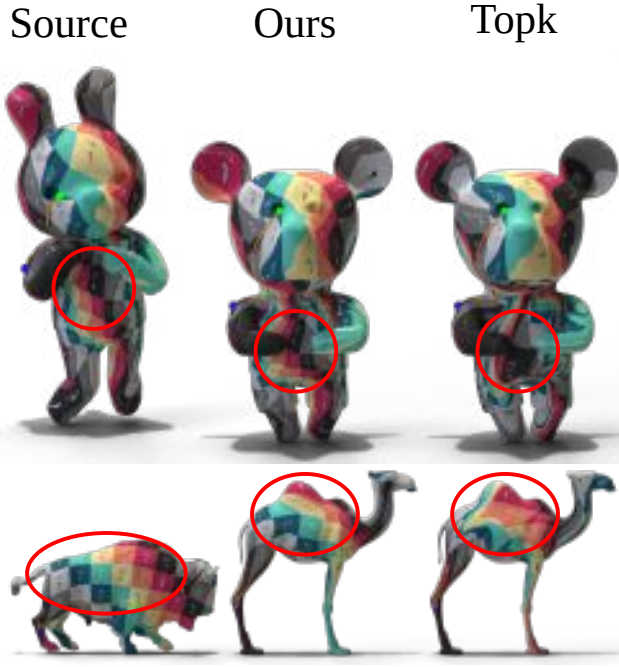
Source   Ours   Topk

Figure 15: ***Scan to SMPL:*** *we first close holes in the raw scan (left) with Meshlab [CCC*08], then we map it onto the template SMPL model [LMR*15] and mask out the surfaced introduced to fill holes. Colored landmarks and paths show* automatically *selected cones and cuts by our method.*

Figure 14: ***Similarity scores. Right****: a map optimized with the top* $k = 100$ *correspondences based on the similarity score.* ***Middle****: map optimized using all correspondences.* ***Left****: source mesh. The map optimized with matches with the highest similarity score matches shows several incorrectness, highlighted with a red circle. This is the result of several incorrect matches which bias the map towards an incorrect energy minimum. Differently, using all correspondences prevents this behavior, as the optimization process automatically filters out wrong matches.*

one mesh onto the other. Finally, the geodesic distance is computed on the target mesh between the centroid of the mapped face to the centroid of the ground truth target face.

In general, DinoV2 [ODM*23] outperforms its predecessor V1 [CTM*21], offering more accurate and robust matches. The depth at which features are extracted (9 vs 11) does not impact the matches of DinoV2, while it plays a significant role for DinoV1, as discussed in [AGBD21]. The presence of texture is beneficial to DinoV1, while it only offers a minor improvement for DinoV2. This is reassuring as our method can only assume access to untextured models. The choice of colored lights offers additional shading and visual features for DinoV1, but it is less relevant for DinoV2 as white lights perform equally with the base case.

Effect of Initial Alignment

We ablate the effect and robustness to misalignment for correspondences quality, see Figure 11. We start from a correct alignment with 12 shape pairs and incrementally misalign one shape - step of $20°$ around the up axis. We report the quality of correspondences in terms of geodesic error, i.e., accuracy. The quality sensibly decreases with severe misalignment - more than $40°$ - reaching a peak with opposite orientation - $180°$. We additionally compare the qual-
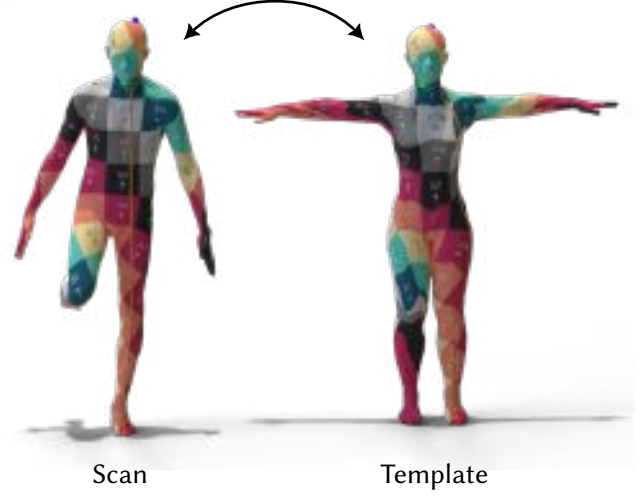
ity of correspondences for the last two layers of Dino-ViT and show that, for such a case, a deeper level (L11) seems to encode slightly better semantic information than the previous layer (L9).

Handling Noise and Holes

Raw scans present noise or holes, thus inhibiting the applicability of our method since it assumes watertight genus zero meshes. Intuitively the presence of large holes, and missing limbs such as arms, may severely mislead DinoViT and thus our pipeline. On the other hand, small holes can be dealt with by applying a simple hole-filling approach. In Figure 15, we use our method to map a raw scan to the SMPL template [LMR*15]. We prefill small holes with Meshlab [CCC*08] and then apply our pipeline.