

EchoMatch: Partial-to-Partial Shape Matching via Correspondence Reflection

Yizheng Xie^{1*} Viktoria Ehm^{1,2*} Paul Roetzer³ Nafie El Amrani³

Maolin Gao^{1,2} Florian Bernard³ Daniel Cremers^{1,2}

¹ Technical University of Munich ² Munich Center for Machine Learning ³ University of Bonn

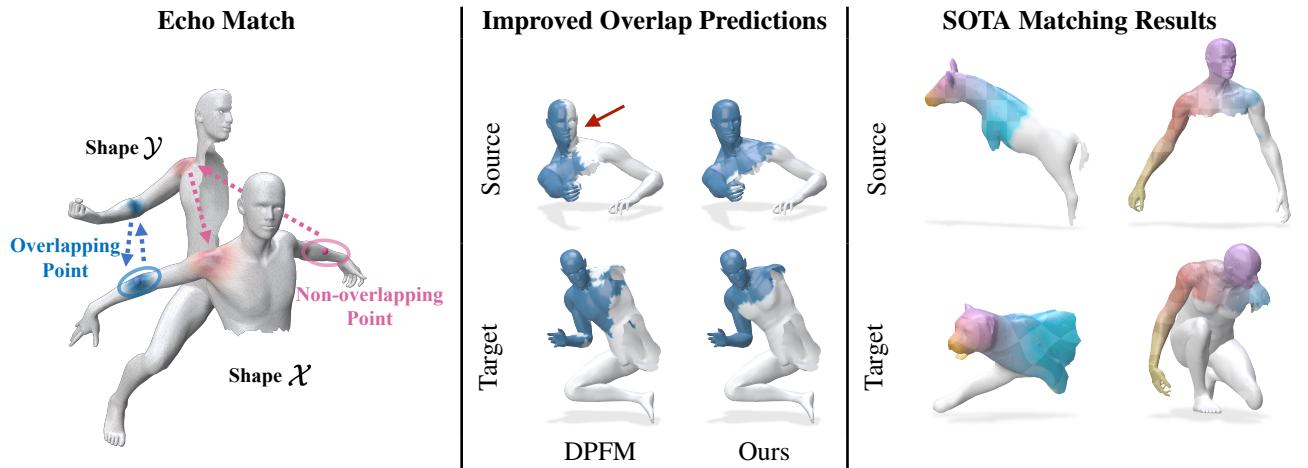


Figure 1. **Left:** We present EchoMatch, a novel framework for partial-to-partial shape matching. We formalise a correspondence reflection approach, where we ensure forward-backward-consistency for overlapping points. If a point returns to its original neighbourhood after sending it from shape \mathcal{X} to shape \mathcal{Y} and back to shape \mathcal{X} , we consider it to be an overlapping point, otherwise it is a non-overlapping point. **Middle:** We show improved overlap prediction for partial-to-partial shape matching compared to the previous SOTA method DPFM [3]. **Right:** We achieve SOTA matching results in the partial-to-partial setting.

Abstract

Finding correspondences between 3D shapes is a crucial problem in computer vision and graphics. While most research has focused on finding correspondences in settings where at least one of the shapes is complete, the realm of partial-to-partial shape matching remains under-explored. Yet, it is important since in many applications shapes are only observed partially due to occlusion or scanning. Finding correspondences between partial shapes comes with an additional challenge: We not only want to identify correspondences between points on either shape but also have to determine which points of each shape actually have a partner. To tackle this challenging problem, we present EchoMatch, a novel framework for partial-to-partial shape matching that incorporates the concept of correspondence reflection to enable an overlap prediction within a func-

tional map framework. With this approach, we show that we can outperform current SOTA methods in challenging partial-to-partial shape matching problems. Our code is publicly available at <https://echo-match.github.io>.

1. Introduction

Shape matching is a long-standing problem in computer vision and beyond and is a fundamental ingredient in addressing tasks such as shape interpolation, animation, or texture transfer. In 3D shape matching, we are interested in finding correspondences between two 3D shapes. There exists extensive research on full-to-full shape matching, where the complete geometry of both shapes is available [1, 11, 25, 43, 62]. The more realistic – but substantially more challenging – scenario is partial shape matching, where either one or both shapes are only observed partially, e.g., due to scanning artefacts or occlusions.

The partial-to-partial matching case is challenging, as we

* These authors contributed equally.

{yizheng.xie, viktoria.ehm}@tum.de

need not only to find correspondences between the shapes, but also to determine the overlapping region between both shapes, either explicitly or implicitly. Only a few methods exist that tackle such partial-to-partial shape matching problems. Among them are two axiomatic methods, SMCOMB [54] and GC-PPSM [24], which are dependent on good feature initialisation and are rather slow, as well as the learning-based method DPFM [3] that struggles with good overlap prediction in challenging settings. In our work, we improve this overlap predictor by introducing EchoMatch. EchoMatch models the matching of shape \mathcal{X} to shape \mathcal{Y} and the matching of shape \mathcal{Y} to shape \mathcal{X} separately. By composing these individual directional mappings, any overlapping point in one shape will return to a neighbourhood close to its original location, much like an echo returning in audio processing. However, a counterpart does not exist in the other shape for non-overlapping points, so these points will be mapped far apart (see left image in Figure 1). With EchoMatch, we improve the overlap prediction and the overall correspondence quality.

As shown in [4, 23], current methods use spatial coordinate features as input features, which makes them dependent on the alignment of shapes. Recently, an increasing number of computer vision methods have based their features on foundation models, such as 2D image features. Inspired by Diff3f [20], which uses DINoV2 [41] features for shape matching, we propose to use image-based features also for partial-to-partial shape matching. With our new EchoMatch formulation, we set the new SOTA in the partial-to-partial shape matching domain with both the commonly used spatial and image-based features. We summarise our contributions as follows:

- We formulate a new overlap predictor for shape matching that checks whether a point would return to its neighbourhood after forward-backward matching.
- We present EchoMatch, a novel shape matching framework that embeds this new overlap predictor in a functional map framework for the challenging partial-to-partial problem.
- We outperform current SOTA methods in partial-to-partial shape matching by a large margin.

2. Related Work

In this section, we summarise the most relevant works to our approach. This includes various shape matching methods, the modelling of neighbourhood relations in matching methods and feature computation for partial shapes.

2.1. Shape Matching

Shape matching can be divided into different categories w.r.t. partiality of shapes. In this sense, we discuss relevant works by starting with methods falling into the full-to-full shape matching category, continuing with partial-to-

full approaches and concluding with partial-to-partial shape matching, the category our approach falls into. For an in-depth review of the shape matching literature, we refer the reader to surveys [17, 55].

Full-to-Full Shape Matching. Non-rigid shape matching methods can be categorised into two groups, namely spatial and spectral methods, depending on their operating domain. In the family of spatial methods, the most common strategy is to deform the source to resemble the target shape and the correspondences are retrieved simply by nearest neighbour search in the embedding space [25, 26, 59]. This is often dubbed non-rigid Iterative Closest Point (ICP) and typically is computationally costly. Recent efforts try to tackle deformation and matching tasks jointly with optimality guarantees. However, the size of tractable problems is limited: only either a handful of sparse key points or low-resolution shapes are matched [29, 53, 54, 63, 64]. On the other hand, spectral methods (e.g. functional maps) pioneered by Ovsjanikov et al. [43] do not depend on the shape resolution but rather the number of spectral bases, which are typically the $50 \sim 200$ leading eigen-bases of the Laplace-Beltrami-Operator (LBO) of the given shapes. Moreover, the resulting optimisation becomes a least squares problem, which can be efficiently solved to optimality. Many learning-based methods built on this idea and with that yield state-of-the-art results [6, 11, 30, 37]. Following these successful works, our approach uses functional maps as regularisation during training.

Partial-to-Full Shape Matching. In the partial-to-full shape matching setting a partial shape is matched to a full shape. This is considerably more difficult than matching two full shapes due to more ambiguities introduced by the partial shape. From an optimisation aspect, this means that we are not trying to find a permutation matrix anymore but rather a partial permutation matrix and thus the search space is dramatically expanded. Rodolà et al. tackle partiality in the spectral space [38, 52] by leveraging the insight that the eigenbases of partial shapes do appear in the eigenbases of the full ones. It lays the cornerstone of a line of deep learning works [3, 11], which achieve impressive results in partial-to-full shape matching setting. The idea of geometric consistency has also been exploited, where neighbouring parts in the source shape must remain neighbours on the target shape [22, 54]. Similar to these methods, we include neighbourhood relations into our framework. Yet, we do not enforce geometric consistency as a hard constraint but rather use neighbourhood relations as a soft regularisation for overlap prediction.

Partial-to-Partial Shape Matching. In the partial-to-partial shape matching setting, we not only need to find correspondences between two partial shapes but also need to find their overlapping region. Even though partial-to-partial shape matching poses the most realistic setting in shape

matching, it is still under-explored, and only three methods tackle this problem: DPFM [3] is a supervised learning method with a cross-attention feature refinement network to allow communication between the learned features. The method struggles mostly with the overlap prediction quality, a problem we improve upon in this paper. The other two methods, SM-COMB [54] and GC-PPSM [24], are axiomatic methods. Both use integer linear programming and geometric consistency constraints to tackle the partial shape matching problem. SM-COMB requires inputs to have no boundary and thus holes of partial shapes need to be closed beforehand. In contrast, GC-PPSM can handle shapes with boundaries. Yet, both methods rely on highly descriptive learned features from [11] to achieve SOTA results. In addition, both methods require large computation times and thus are not scalable to higher resolutions. In contrast, our approach is fast, depends less on feature initialisation and is scalable.

2.2. Neighbourhoods in Matching

Two different types of mechanisms to model neighbourhood preservation exist in literature, that are most relevant to our proposed shape matching framework. The first mechanism is Mutual Nearest Neighbors (MNN), or bilateral neighbours. Two points of two distinct sets are MNN if they are each other's nearest neighbour. MNN is widely used in computer vision applications, such as point cloud matching [31, 42]. Furthermore, MNN can be interpreted as a special case of cycle-consistent matching between two shapes and was applied in many shape matching frameworks [7, 32, 44]. The second mechanism is Mutual Nearest Neighbors Consensus (MNNC). Compared to MNN, MNNC additionally incorporates neighbourhood relationships between two sets of points and within a set of points. Overall, MNN and MNNC have been applied across various applications including image matching [13, 35, 48–50, 58], shape retrieval [45], graph matching [27], and point cloud alignment [36, 39, 46, 65]. In particular, for the task of shape matching, Cao et al. [12] extend these concepts by diffusing input functions on corresponding points on both shapes to ensure smooth pointwise correspondences.

MNN can be viewed as a special case of our approach when we limit neighbourhood size to 1. Yet, plainly computing agreeing matchings from \mathcal{X} to \mathcal{Y} , and \mathcal{Y} to \mathcal{X} are not typically well-suited for overlap region prediction since the lack of explicit modelling of the neighbourhood could lead to little overlaps, a gap that we address in our work.

2.3. Feature Descriptors for Partial Shape Matching

Shape descriptors are an essential input for partial shape matching methods, as they serve as the basis for the correspondence prediction. While there exist multiple intrinsic descriptors for full shapes, such as the Wave Kernel Signa-

ture (WKS) [5] or the Heat Kernel Signature (HKS) [57], they are not applicable for partial shapes. Primarily, extrinsic feature descriptors like SHOT [60], based on spatial coordinates and surface normals, or spatial coordinates directly, are used for partial shape matching. Recently, more computer vision methods have used image-based features of foundation models. In [61], the authors project a 3D mesh to 2D images to compute 2D labels and project these labels back to 3D. Aggregating 2D foundation model features to 3D has already shown significant results [16, 28, 34]. Diff3f [20] utilises this idea for shape matching and uses DINOv2 [41] features on the rendered images. Concurrent work [14] uses these image features for partial point cloud matching. We show that these features can benefit partial-to-partial shape matching.

3. Background

In the following, we summarise the most relevant background for our work. This includes definition of shapes and feature extractors, discussion of functional maps as well as discussion of useful loss formulations.

Definitions. We define two shapes \mathcal{X}, \mathcal{Y} as tuples $\mathcal{X} = (V_{\mathcal{X}}, T_{\mathcal{X}})$ and $\mathcal{Y} = (V_{\mathcal{Y}}, T_{\mathcal{Y}})$, with respective vertices $V_{\mathcal{X}}$ and $V_{\mathcal{Y}}$ and respective triangles $T_{\mathcal{X}}$ and $T_{\mathcal{Y}}$. We denote the number of vertices per shape as $m = |V_{\mathcal{X}}|$ and $n = |V_{\mathcal{Y}}|$. The first k eigenfunctions of the Laplacians of the shapes are denoted as $\Phi_{\mathcal{X}} \in \mathbb{R}^{m \times k}$ and $\Phi_{\mathcal{Y}} \in \mathbb{R}^{n \times k}$, describing the spectral embedding of the two shapes. We utilise a functional map module for correspondence prediction and a commonly used feature extractor in our network.

Feature Extractor. On both shapes, we learn d -dimensional features $\mathcal{F}_{\mathcal{X}} \in \mathbb{R}^{m \times d}$ and $\mathcal{F}_{\mathcal{Y}} \in \mathbb{R}^{n \times d}$. These features are learned in a Siamese fashion [11], i.e., the same feature extractor is used for both shapes \mathcal{X} and \mathcal{Y} [3, 11, 18].

Full-to-Full Functional Maps. We utilise the regularised functional map solver [19] to compute functional maps. We project the feature descriptors onto their respective spectral projectors, formed by the pseudoinverse of the first k eigenfunctions of the Laplace-Beltrami basis. This results in the following projections:

$$\mathbf{A} = \Phi_{\mathcal{X}}^{\dagger} \mathcal{F}_{\mathcal{X}} \quad \text{and} \quad \mathbf{B} = \Phi_{\mathcal{Y}}^{\dagger} \mathcal{F}_{\mathcal{Y}},$$

where $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ represent the spectral bases on \mathcal{X} and \mathcal{Y} , respectively, while $\Phi_{\mathcal{X}}^{\dagger}$ and $\Phi_{\mathcal{Y}}^{\dagger}$ denote their corresponding pseudoinverses. In the full-to-full functional map pipeline, we seek to find a functional map $C_{\mathcal{XY}}$ that aligns shapes \mathcal{X} and \mathcal{Y} in the spectral domain. With the derived coefficients, we apply regularisation techniques, such as commutativity constraints [19]. Specifically, we solve for $C_{\mathcal{XY}}$ as follows:

$$C_{\mathcal{XY}} = \arg \min_C \|CA - B\|_F^2 + \lambda \|C\Delta_{\mathcal{X}} - \Delta_{\mathcal{Y}}C\|_F^2.$$

Here $\Delta_{\mathcal{X}}$ and $\Delta_{\mathcal{Y}}$ denotes the resolvent operator for commutativity regularisation [47] and λ is a scalar hyperpa-

parameter. An analogous formalism was also applied to the partial-to-full setting [3, 52].

Partial-to-Partial Functional Map. As demonstrated in [3, 9], for both full-to-partial and partial-to-partial shape matching, directly applying the formulation from the previous section can introduce errors in the functional map. We address this issue through feature pruning. In our training setup, we denote the ground truth vertex overlap mask by M_Y , which isolates the region of \mathcal{Y} that overlaps with \mathcal{X} . These masks enable us to restrict feature descriptors to the overlapping region, focusing the functional map on shared areas only. To compute the spectral coefficients of the feature descriptors on each shape, we project \mathcal{F}_X and the masked feature descriptors $M_Y \odot \mathcal{F}_Y$ onto their respective spectral bases where \odot denotes element-wise multiplication. This gives

$$\mathbf{A} = \Phi_X^\dagger \mathcal{F}_X \quad \text{and} \quad \mathbf{B} = \Phi_Y^\dagger (M_Y \odot \mathcal{F}_Y).$$

By retaining only the ground truth overlapping region, the resulting functional map can be interpreted as a partial-to-full functional map. We mask only the overlapping parts of \mathcal{Y} (denoted $\bar{\mathcal{Y}}$), ensuring that all regions in $\bar{\mathcal{Y}}$ are present in \mathcal{X} . While features on \mathcal{Y} are pruned, features on \mathcal{X} remain unpruned to maintain a stable solution for $\|C\mathbf{A} - \mathbf{B}\|$. Pruning features on \mathcal{X} could lead to a rank-deficient \mathbf{A} , particularly in cases with minimal overlap. Mapping in the reverse direction follows a similar formulation. Experimental analyses can be found in the ablation studies section.

Functional Map Loss. For the functional map module, we compare the predicted functional map $C_{\mathcal{X}\mathcal{Y}}$ with the ground truth functional map C_{gt} , i.e.

$$L_{\text{fmap}} = \|C_{gt} - C_{\mathcal{X}\mathcal{Y}}\|_F^2. \quad (1)$$

PointInfoNCE Loss. Further, we utilise the PointInfoNCE Loss to ensure that the feature distance between corresponding points is minimised, whereas the distance of non-corresponding points is maximised. Refer to [3] for additional details.

4. Our EchoMatch Method

For two shapes \mathcal{X} and \mathcal{Y} , we learn features for each vertex on these shapes. These features are used to compute the functional map and to determine the overlap prediction, see Figure 2. Inspired by point cloud matching [36, 65], EchoMatch is based on a simple yet efficient idea: We model the matching in each direction separately. Each direction is treated as a partial-to-full matching, where every vertex in the partial shape has a counterpart in the (virtual) full shape. This allows us to represent the correspondences as a row-stochastic matrix (see **Soft Point Map** below). After composing the directional mapping from \mathcal{X} to \mathcal{Y} and from \mathcal{Y} to \mathcal{X} , we expect that an overlapping point in shape

\mathcal{X} will return to a neighbourhood close to its original spatial position, see Figure 3. In contrast, for a point in shape \mathcal{X} that is not in the overlapping region, there is no true counterpart in shape \mathcal{Y} due to partiality. Consequently, this point will be mapped to an incorrect location on shape \mathcal{Y} and will likely be mapped back to incorrect areas on shape \mathcal{X} upon returning. In the following section, we will explain EchoMatch in more detail.

Soft Point Map. After extracting features from the feature extractor, they can be used to generate a row-stochastic pointwise correspondence map $\Pi_{\mathcal{X}\mathcal{Y}}$, where the entry $\Pi_{\mathcal{X}\mathcal{Y}}[i, j]$ in i -th row and j -th column of $\Pi_{\mathcal{X}\mathcal{Y}}$ encodes the probability that the vertex $v_i \in V_{\mathcal{X}}$ is mapped to the vertex $v_j \in V_{\mathcal{Y}}$. This correspondence is generated using the softmax function, such that

$$\Pi_{\mathcal{X}\mathcal{Y}} = \text{Softmax}(\mathcal{F}_{\mathcal{X}} \mathcal{F}_{\mathcal{Y}}^T / \tau). \quad (2)$$

Here τ is a temperature parameter controlling the smoothness of the correspondence probabilities.

Correspondence Reflection. The two soft correspondence matrices $\Pi_{\mathcal{X}\mathcal{Y}} \in \mathbb{R}^{m \times n}$ and $\Pi_{\mathcal{Y}\mathcal{X}} \in \mathbb{R}^{n \times m}$ are both row-stochastic, meaning that each row sums to 1. These entries can also be used as overlap predictors by projecting a point from shape \mathcal{X} to shape \mathcal{Y} and back to shape \mathcal{X} via

$$P_{\mathcal{X}} = \Pi_{\mathcal{X}\mathcal{Y}} \times \Pi_{\mathcal{Y}\mathcal{X}}. \quad (3)$$

The diagonal of this matrix encodes the probability that a point on shape \mathcal{X} maps back to itself after being projected onto shape \mathcal{Y} and then returned to \mathcal{X} . For a diagonal element, this probability is given by

$$P_{\mathcal{X}}^{ii} = \Pi_{\mathcal{X}\mathcal{Y}}[i, :] \times \Pi_{\mathcal{Y}\mathcal{X}}[:, i]. \quad (4)$$

Here “ $:$ ” means selecting the whole row (column, respectively) of $\Pi_{\mathcal{X}\mathcal{Y}}$ or $\Pi_{\mathcal{Y}\mathcal{X}}$.

The value $P_{\mathcal{X}}^{ij}$ represents the probability of vertex v_i returning to a vertex $v_j \in \mathcal{X}$ after mapping from shape \mathcal{X} to \mathcal{Y} and back to \mathcal{X} , i.e.

$$P_{\mathcal{X}}^{ij} = \Pi_{\mathcal{X}\mathcal{Y}}[i, :] \times \Pi_{\mathcal{Y}\mathcal{X}}[:, j]. \quad (5)$$

Let N_i denote the neighbourhood of vertex v_i on shape \mathcal{X} with neighbourhood size $l = |N_i|$. For every vertex v_i we stack all probabilities $P_{\mathcal{X}}^{ij}$ for $j \in N_i$ to get $P_{\mathcal{X}}^{N_i} \in \mathbb{R}^{1 \times l}$.

The matrix $P_{\mathcal{X}}^N \in \mathbb{R}^{m \times l}$ is build up from the individual $P_{\mathcal{X}}^{N_i} \in \mathbb{R}^{1 \times l}$ and encodes the probability for each vertex \mathcal{X} to return to its respective neighbourhood, a concept that we refer to as *correspondence reflection*. It indicates whether respective points are in the overlap region or not.

Diffusion for Smoothing. We feed the reflection scores from each point’s neighbourhood $P_{\mathcal{X}}^N$ into a Diffusion Net [56], which further refines the results, producing prediction vectors $p_{\mathcal{X}} \in [0, 1]^m$ and $p_{\mathcal{Y}} \in [0, 1]^n$ for shapes

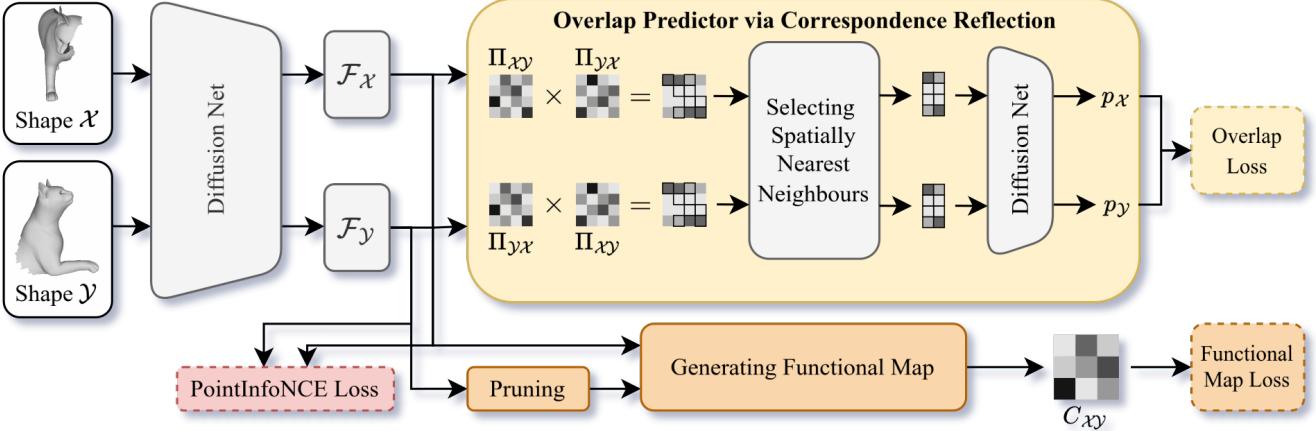


Figure 2. The **EchoMatch** pipeline consists of four main parts: 1) For two shapes \mathcal{X}, \mathcal{Y} , we extract per-vertex feature vectors (collected in feature matrices $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$) using a Diffusion Net [56]. 2) Using these features, we predict an overlap score for every point utilising EchoMatch. 3) We compute the functional map utilising the features (we prune $\mathcal{F}_\mathcal{Y}$ to this end). 4) Finally, we calculate a point-wise normalised cross-entropy (PointInfoNCE) loss for both feature matrices $\mathcal{F}_\mathcal{X}$ and $\mathcal{F}_\mathcal{Y}$.



Figure 3. **Correspondence Reflection:** Starting from a query point on shape \mathcal{X} with a corresponding point on shape \mathcal{Y} (both in black), we propagate soft probability scores forth and back using the soft point maps $\Pi_{\mathcal{X}\mathcal{Y}}$ and $\Pi_{\mathcal{Y}\mathcal{X}}$. This reflection yields “echoed” scores on \mathcal{X} , which are further refined after aggregating the spatial neighbourhood of the original query point (highlighted in light blue).

\mathcal{X} and \mathcal{Y} , respectively. Using the Diffusion Net, the model learns to dynamically adjust the influence of neighbouring values during training, effectively smoothing out symmetry artefacts and spatial inconsistencies. This refinement step leverages both local neighbourhood information and global surface learning to achieve more coherent overlap predictions. We show a refinement example in Figure 4.

Overlap Loss. Given the ground truth overlap region $M_\mathcal{X} \in \{0, 1\}^m$ and $M_\mathcal{Y} \in \{0, 1\}^n$ and the predictions $p_\mathcal{X} \in [0, 1]^m$ and $p_\mathcal{Y} \in [0, 1]^n$ we define the overlap loss as

$$L_{ov} = \text{wBCE}(M_\mathcal{X}, p_\mathcal{X}) + \text{wBCE}(M_\mathcal{Y}, p_\mathcal{Y}). \quad (6)$$

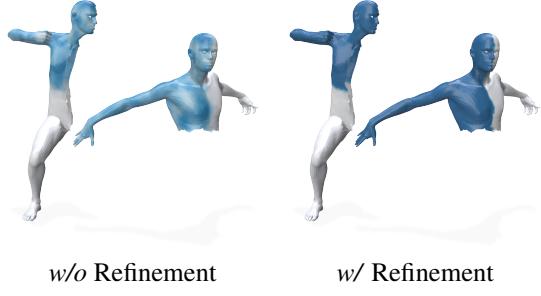


Figure 4. **Refinement of Overlap Predictions:** Before refinement with Diffusion Net, the reflection scores (summed across the neighbourhood for visualisation) appear fuzzy and irregular (left). After refinement, we obtain a clear and well-defined final overlap prediction (right).

5. Experimental Results

We experimentally evaluate our proposed method and refer to the supplementary material for implementation details.

5.1. Partial Shape Matching Datasets

We use all three existing partial-to-partial shape matching datasets for our experiments. Cuts-Partial-to-Partial (CP2P24) [3, 24], based on SHREC’16 [15] and TOSCA [10] includes isometric, normalised, aligned humanoid and animal shapes. PARTIALSMAL (PS-MAL) [24] based on SMAL [66], includes non-isometric, normalised animal shapes. Additionally, we use the most challenging dataset BeCoS [23], based on multiple datasets [2, 8, 10, 21, 40, 51, 51, 66]. It includes non-isometric, realistic scaled humanoid and animal shapes.

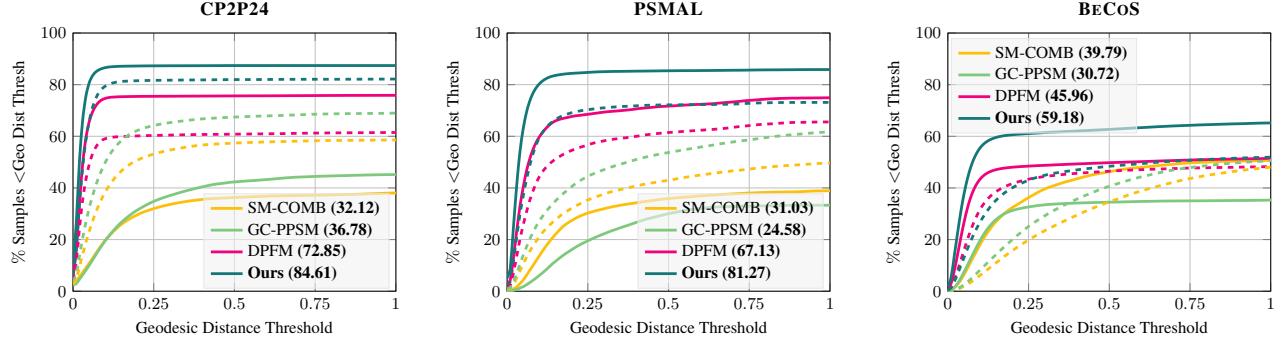


Figure 5. We show **PCK curves** and respective AUC (\uparrow) values (numbers in legends) for the different partial-to-partial shape matching methods on three different datasets using DINOv2 features (solid) and, for completeness, using spatial coordinates as inputs (dashed).

5.2. Shape Matching Methods

Even though partial-to-partial shape matching is the most realistic in the shape matching domain, it is still under-explored. There exists only one supervised learning-based method: DPFM [3] and two axiomatic methods: SM-COMB [54] and GC-PPSM [24]. As SM-COMB can only solve shape matching problems that include closed shapes, we add vertices to close the shapes beforehand. Matchings that include these added vertices later are evaluated as non-overlapping regions. We use a common number of training iterations and select the last checkpoint for both learning-based methods (DPFM and our method), ensuring consistency across evaluations, except for BeCoS, which has designated validation and test splits.

5.3. Input Features

We evaluate our method on both the previously used spatial coordinates as input and foundation model image features projected on 3D shapes.

Coordinates as Input. Previous partial shape matching methods mostly used spatial coordinates (XYZ) input for their methods. DPFM uses XYZ coordinates as direct input with only a few augmentations. SM-COMB and GC-PPSM use features from a strong network pre-trained on the unsupervised partial-to-full shape matching task with XYZ-coordinates as input [11], assuming spatial alignment and full-shape priors.

Image Features as Input. In [4, 23], the authors already mention that due to spatial coordinates as input features, current methods mainly depend on aligned shapes. Recently, more computer vision methods rely on features from foundation models, e.g., 2D image features. These features are not directly applicable for partial-to-partial shape matching as they cannot determine overlapping regions and output noisy correspondences. However, we show that the image features (DINOv2 [41] features) benefit the performance of our method as input features and exceed the

performance of the XYZ input features, especially when the shapes are not aligned. We use the framework from Diff3f [20] to extract the image features from the mesh by rotating a camera around the mesh from 100 different views. Compared to Diff3f, we only use the computed image features and do not include diffusion features, as we do not know which animal we observe. Additionally, we use back-face culling since we observe partial shapes. For fairness, we use DINOv2 features directly for all methods.

5.4. Evaluation Metric

Intersection Over Union. As proposed in [24], we use the Intersection over Union (IoU) to evaluate the overlap region prediction. The prediction $P \in \{0, 1\}^{|V| \times 1}$ and the ground truth vector $G \in \{0, 1\}^{|V| \times 1}$ encode for every vertex in V if it is in the overlapping region. Then, we get the quality of the overlapping region with the IoU = $\frac{|P \cap G|}{|P \cup G|}$. We report the mean IoU (mIoU) over all shapes.

Geodesic Error. We evaluate correspondences with the well-established geodesic error using the Princeton Protocol [33]. We base our evaluation on the partial-to-partial evaluation of [23]: We normalise the geodesic error by the shape diameter (square root of the area) of the corresponding full shape. For unmatched vertices that should be matched and vertices that are matched but should not be matched, we set an infinite geodesic error [24]. This ensures that we incorporate the overlap prediction in the geodesic error computation and helps to get a better picture of the overall performance. Note that due to this infinite geodesic error, the geodesic error curve does not necessarily reach 100%. When referring to the mean geodesic error, we evaluate the geodesic error only in the overlapping region.

5.5. Results

We show the geodesic error curves with Area Under the Curves (AUC) in Figure 5, the mean Intersection over Union (mIoU) in Table 1 and some qualitative results with DINOv2 features in Figure 6. We refer to the supplemen-

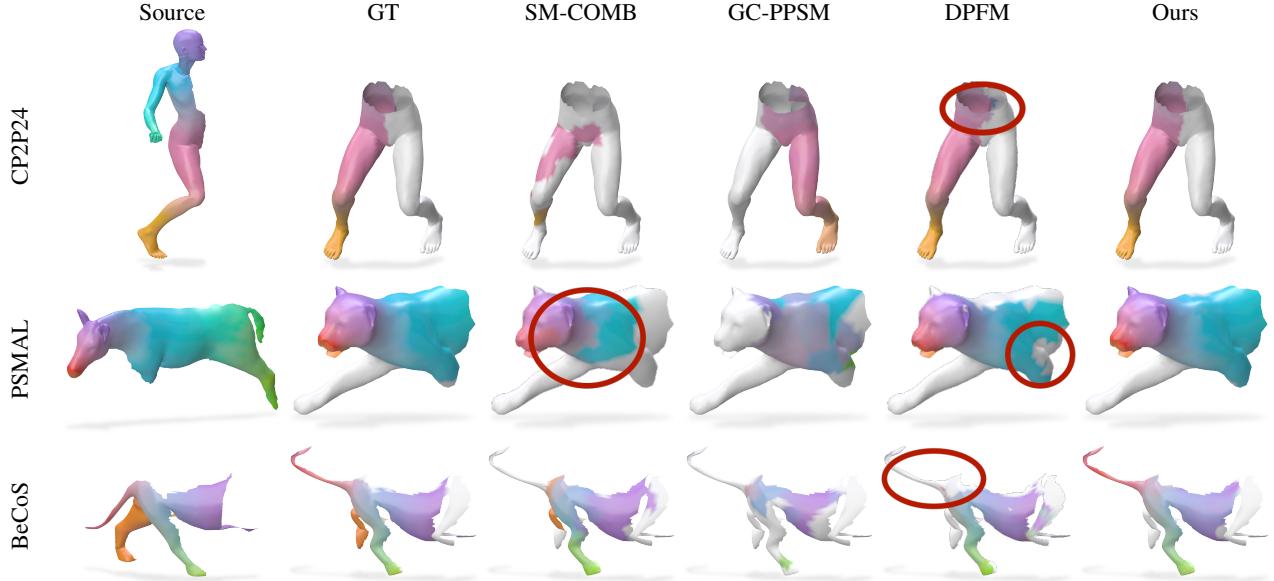


Figure 6. We show **qualitative results** of all methods used in our experiments on datasets BeCoS, PSMAL and CP2P24. The axiomatic methods SM-COMB [54] and GC-PPSM [24] struggle with capturing the full extent of the overlapping region as well as producing smooth matchings. DPFM [3] generally performs well, however, its overlap predictions can be scattered around the boundaries of the overlapping regions. In contrast, our method **EchoMatch**, produces well-defined overlap predictions and smooth matchings across all benchmarks.

Methods	CP2P24 [3, 24]	PSMAL [24]	BeCoS [23]
SM-COMB [54]	XYZ	57.86*	54.76*
	DINOv2	38.38	47.04
GC-PPSM [24]	XYZ	69.29*	49.34
	DINOv2	49.66	34.30
DPFM [3]	XYZ	63.86	48.18
	DINOv2	74.15	51.02
Ours	XYZ	80.10	52.40
	DINOv2	84.72	84.75

Table 1. **Mean IoU ($\times 100$) on different datasets:** Our method outperforms previous methods in terms of the mean Intersection over Union in all available partial-to-partial shape matching datasets, especially in the challenging non-isometric BeCoS dataset. Some axiomatic methods (see “*”) use features computed with a pre-trained network trained on partial-to-full shapes.

plementary material for analysis of incompleteness patterns A.1, full curves A.2 and more qualitative results A.5.

Discussion. Both axiomatic methods rely heavily on good input features and perform mostly well when utilising learned features. Both solve their problem in low dimension, such that they suffer from patch-wise artefacts when upsampling to a higher dimension (see Figure 6: GC-PPSM and SM-COMB in PSMAL). DPFM mostly struggles with good overlap predictions (see Figure 6 BeCoS). In terms of mIoU, our method outperforms current methods in both XYZ and DINOv2 features, as shown in Table 1.

In terms of geodesic error (see Figure 6) curves, we outperform previous state-of-the-art on all datasets and metrics, except on BeCoS using spatial features. Here, large performance drops for all methods stem from the dataset’s inherent challenges (shapes are unaligned, unnormalised, and include challenging non-isometries), which make spatial features fundamentally ill-suited for evaluated methods and yield near-random performance for all methods. As a consequence, results cannot be interpreted as one method outperforming another. Yet, when using DINOv2 features, our method exceeds all other methods by a large margin.

5.6. Ablation Studies

We show ablation studies on different neighbourhood sizes, pruning with the ground truth overlap, rotation experiments and a discretisation example. We refer to the supplementary material for further ablations on general shape matching and loss formulations, where we show that combining our three loss parts shows the most stable results with two datasets.

Different Neighbourhood Sizes. We test different neighbourhood sizes l on the BeCoS validation set. We show the average geodesic error on the overlapping region and the mIoU in Table 2. A larger neighbourhood significantly improves performance, mainly affecting the overlap predictor in terms of mIoU. Stable performance for $l = 32 \dots 1024$ is due to “Overlap DiffusionNet”, which can flexibly learn the influence of the points in the neighbourhood and can ignore points further away.

Size l	1	4	8	16	32	64	128	256	512	1024
mIoU (\uparrow)	55.04	62.86	57.91	61.64	65.00	67.43	65.10	67.49	67.95	67.81
GeoErr (\downarrow)	7.96	8.03	8.17	7.97	7.86	7.80	8.18	8.11	7.85	8.08

Table 2. **Neighbourhood Ablation:** We show the mean IoU ($\times 100$) and Geodesic Error ($\times 100$) on the BeCoS validation set for different neighbourhood sizes. Larger neighbourhoods perform better than smaller ones, where 64 shows the best results.

	w/ Pruning	w/o Pruning	Pruning Both
mIoU (\uparrow)	67.43	67.30	65.48
GeoErr (\downarrow)	7.80	8.72	8.80

Table 3. By pruning \mathcal{F}_Y before computing the functional map both the **mean IoU and the geodesic error** improve.

Train: Rotate (around)					
	XYZ (small)	Y	Z	XYZ	DINOv2
Test w/o rotation	9.13	10.76	16.71	11.84	7.23
Test w/ rotation	33.34	21.56	27.61	12.36	9.73

Table 4. **Rotation Ablation:** We train with spatial coordinates as input on only little rotated shapes ($XYZ(small)$), with random rotation around Y , Z or all axes. Additionally, we train without rotation with DINOv2 features. Then, we test randomly rotated or aligned shapes, as given in the dataset. When training with low rotation and spatial input features, we get good results on the proposed test set, but adding rotation worsens the results. Using DINOv2 features improves the results compared to spatial inputs in both the aligned test set and the randomly rotated one.

Pruning with Ground Truth Overlap. We show that feature pruning in the functional map module (see Section 2.3 and [3, 9]) with the ground truth overlap region during training enhances performance in the BeCoS [23] validation set in Table 3. Importantly, ground truth overlap is only used during training; our overlap predictor relies solely on the learned feature at test times.

Rotation Around Different Axes. As mentioned in [4, 23], previous partial-to-partial shape matching methods struggle with overfitting to the alignment of datasets. We show an ablation study on the aligned CP2P21 dataset [3]. As a baseline, we rotate the shapes only slightly, as proposed in DPFM [3]. Moreover, following ULRSSM [11], we rotate around the y-axis and z-axis. Additionally, we rotate the shapes around all three axes up to 360 degrees. We also train the network with aligned shapes and DINOv2 features. With these trained models we test the original, aligned dataset and a randomly rotated one. In Table 4 we show the average geodesic error on the overlapping region. All input features are affected by the rotation in the randomly rotated test set. Still, the DINOv2 input features outperform the spatial coordinates by a large margin.

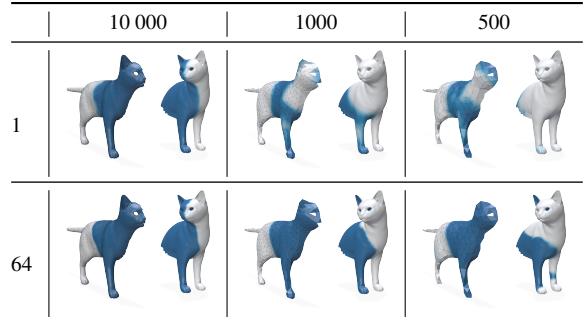


Table 5. **Discretisation Experiment:** We examine the effect of different neighbourhood sizes (1 and 64) of the overlap prediction at different discretisation levels of on one shape (10 000, 1000, and 500 faces). We do not alter the resolution of the other shape. A larger neighbourhood (64) leads to more reliable overlap predictions, even with coarse discretisation. However, errors still occur, as seen in the right column for the left leg of the rightmost shape.

Discretisation Example. We show exemplary that with a larger neighbourhood, our method is more robust to different discretisation levels (Table 5).

6. Limitations and Future Work

We demonstrate that using 2D image features significantly improves matching performance compared to spatial coordinates. Yet, our approach still depends on the quality of these features, which poses certain limitations. For example, points that are structurally part of the mesh but not visible in any image (e.g. the tongue of certain animal shapes) must be approximated, often leading to inaccuracies. Additionally, matching partial shapes introduces challenges since views of the 3D shape may not have meaningful 2D image representations but are still used for feature computation. We observed that the quality of image features for partial shapes is still notably lower than for complete shapes. Finally, although our method effectively handles different discretisations, it is not entirely invariant, see Table 5.

7. Conclusion

We present EchoMatch, a novel framework to address the challenging partial-to-partial shape matching problem. By introducing the concept of correspondence reflection within a functional map framework, EchoMatch effectively identifies overlapping regions between partially observed shapes. This approach yields state-of-the-art performance, setting a new benchmark for realistic applications where shapes are only partially visible. We hope our work inspires further exploration of correspondence reflection in the under-explored domain of partial-to-partial shape matching. In addition, we believe that our overlap predictor can be applied to other problems in visual computing that deal with partial data.

Acknowledgments

We thank Dongliang Cao, Lennart Bastian and Daniel Scholz for helpful discussions. This work was supported by the ERC Advanced Grant “SIMULACRON” (agreement #884679) and the ERC Starting Grant “Harmony” (#101160648).

References

- [1] Ahmed Abdelreheem, Abdelrahman Eldekokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023. 1
- [2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM SIGGRAPH 2005 Papers*, pages 408–416, 2005. 5
- [3] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *3DV*, pages 175–185. IEEE, 2021. 1, 2, 3, 4, 5, 6, 7, 8
- [4] Souhaib Attaiki, Lei Li, and Maks Ovsjanikov. Generalizable local feature pre-training for deformable shape analysis. In *CVPR*, 2023. 2, 6, 8
- [5] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, pages 1626–1633, 2011. 3
- [6] Lennart Bastian, Yizheng Xie, Nassir Navab, and Zorah Lähner. Hybrid functional maps for crease-aware non-isometric shape matching. *CVPR*, 2024. 2
- [7] Florian Bernard, Daniel Cremers, and Johan Thunberg. Sparse quadratic optimisation over the stiefel manifold with application to permutation synchronisation. *Advances in Neural Information Processing Systems*, 34:25256–25266, 2021. 3
- [8] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *CVPR*, pages 3794–3801, 2014. 5
- [9] Amit Bracha, Thomas Dagès, and Ron Kimmel. On unsupervised partial shape correspondence. In *ACCV*, pages 4488–4504, 2024. 4, 8
- [10] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008. 5, 1
- [11] Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *ACM Transactions on Graphics (TOG)*, 2023. 1, 2, 3, 6, 8
- [12] Dongliang Cao, Zorah Laehner, and Florian Bernard. Synchronous diffusion for unsupervised smooth non-rigid 3d shape matching. In *ECCV*, 2024. 3
- [13] Jan Cech, Jiri Matas, and Michal Perdoch. Efficient sequential correspondence selection by cosegmentation. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1568–1581, 2010. 3
- [14] Zhangquan Chen, Puhua Jiang, and Ruqi Huang. Unsupervised non-rigid point cloud matching through large vision models. *arXiv preprint arXiv:2408.08568*, 2024. 3
- [15] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, Y Sahillioglu, et al. Shrec’16: Partial matching of deformable shapes. *Proc. 3DOR*, 2(9):12, 2016. 5, 1
- [16] Dale Decatur, Itai Lang, and Rana Hanocka. 3d highlighter: Localizing regions on 3d shapes via text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20930–20939, 2023. 3
- [17] Bailin Deng, Yuxin Yao, Roberto M. Dyke, and Juyong Zhang. A Survey of Non-Rigid 3D Registration. *Computer Graphics Forum*, 41(2):559–589, 2022. *leprint*: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14502>. 2
- [18] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, pages 8592–8601, 2020. 3
- [19] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric maps: Robust feature learning for shape correspondence. *CVPR*, 2020. 3
- [20] Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *CVPR*, pages 4494–4504, 2024. 2, 3, 6
- [21] Roberto M Dyke, Yu-Kun Lai, Paul L Rosin, Stefano Zappana, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. Shrec’20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020. 5
- [22] Viktoria Ehm, Paul Roetzer, Marvin Eisenberger, Maolin Gao, Florian Bernard, and Daniel Cremers. Geometrically consistent partial shape matching. *3DV*, 2023. 2
- [23] Viktoria Ehm, Nafie El Amrani, Yizheng Xie, Lennart Bastian, Maolin Gao, Weikang Wang, Lu Sang, Dongliang Cao, Zorah Lähner, Daniel Cremers, et al. Beyond complete shapes: A quantitative evaluation of 3d shape matching algorithms. *arXiv preprint arXiv:2411.03511*, 2024. 2, 5, 6, 7, 8, 1
- [24] Viktoria Ehm, Maolin Gao, Paul Roetzer, Marvin Eisenberger, Daniel Cremers, and Florian Bernard. Partial-to-partial shape matching with geometric consistency. *CVPR*, 2024. 2, 3, 5, 6, 7, 1
- [25] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020. 1, 2
- [26] Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. Elastic correspondence between triangle meshes. In *Computer Graphics Forum*, pages 121–134. Wiley Online Library, 2019. 2
- [27] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020. 3
- [28] Michael Fischer, Zhengqin Li, Thu Nguyen-Phuoc, Aljaz Bozic, Zhao Dong, Carl Marshall, and Tobias Ritschel. Nerf analogies: Example-based visual attribute transfer for nerfs. In *CVPR*, pages 4640–4650, 2024. 3
- [29] Maolin Gao, Paul Roetzer, Marvin Eisenberger, Zorah Lähner, Michael Moeller, Daniel Cremers, and Florian

- Bernard. SIGMA: Quantum scale-invariant global sparse shape matching. In *ICCV*, 2023. 2
- [30] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, pages 4370–4379, 2019. 2
- [31] Itan Hezroni, Amnon Drory, Raja Giryes, and Shai Avidan. Deepbbs: Deep best buddies for point cloud registration. In *3DV*, pages 342–351. IEEE, 2021. 3
- [32] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Computer graphics forum*, pages 177–186, 2013. 3
- [33] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 6
- [34] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 3
- [35] Shuda Li, Kai Han, Theo W Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *CVPR*, pages 10196–10205, 2020. 3
- [36] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *CVPR*, 2022. 3, 4
- [37] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, pages 5659–5667, 2017. 2
- [38] Or Litany, Emanuele Rodolà, Alexander M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, pages 247–258. Wiley Online Library, 2017. 2
- [39] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *ICCV*, pages 16014–16023, 2021. 3
- [40] Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. Smooth non-rigid shape matching via effective dirichlet energy optimization. In *3DV*, pages 495–504. IEEE, 2022. 5
- [41] Maxime Oquab, Timothée Darcret, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023. 2, 3, 6
- [42] Shaul Oron, Tali Dekel, Tianfan Xue, William T Freeman, and Shai Avidan. Best-buddies similarity—robust template matching using mutual nearest neighbors. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1799–1813, 2017. 3
- [43] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 1, 2
- [44] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. *Advances in neural information processing systems*, 26, 2013. 3
- [45] Vittal Premachandran and Ramakrishna Kakarala. Consensus of k-nns for robust neighborhood selection on graph-based manifolds. In *CVPR*, pages 1594–1601, 2013. 3
- [46] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *CVPR*, pages 11143–11152, 2022. 3
- [47] Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*, pages 39–53. Wiley Online Library, 2019. 3, 1
- [48] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in neural information processing systems*, 31, 2018. 3
- [49] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *ECCV*, pages 605–621. Springer, 2020.
- [50] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1020–1034, 2020. 3
- [51] Emanuele Rodola, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *CVPR*, pages 4177–4184, 2014. 5
- [52] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer graphics forum*, pages 222–236. Wiley Online Library, 2017. 2, 4
- [53] Paul Roetzer and Florian Bernard. Spidermatch: 3d shape matching with global optimality and geometric consistency. In *CVPR*, 2024. 2
- [54] Paul Roetzer, Paul Swoboda, Daniel Cremers, and Florian Bernard. A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *CVPR*, pages 428–438, 2022. 2, 3, 6, 7
- [55] Yusuf Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. 2
- [56] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 4, 5, 1
- [57] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. 3
- [58] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loft: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 3
- [59] Ramana Sundaraman, Nicolas Donati, Simone Melzi, Etienne Cormann, and Maks Ovsjanikov. Deformation recovery:

- Localized learning for detail-preserving deformations. *SIGGRAPH Asia*, 2024. 2
- [60] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *ECCV*, pages 356–369. Springer, 2010. 3
- [61] Yunhai Wang, Minglun Gong, Tianhua Wang, Daniel Cohen-Or, Hao Zhang, and Baoquan Chen. Projective analysis for 3d shape segmentation. *ACM Transactions on Graphics (TOG)*, 32(6):1–12, 2013. 3
- [62] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3d: Few-shot 3d keypoint detection with back-projected 2d features. In *CVPR*, 2024. 1
- [63] Thomas Windheuser, Ulrich Schlickwei, Frank R Schmidt, and Daniel Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *ICCV*, pages 2134–2141. IEEE, 2011. 2
- [64] Thomas Windheuser, Ulrich Schlickwei, Frank R Schmidt, and Daniel Cremers. Large-scale integer linear programming for orientation preserving 3d shape matching. In *Computer Graphics Forum*, pages 1471–1480. Wiley Online Library, 2011. 2
- [65] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *CVPR*, 2023. 3, 4
- [66] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017. 5, 1

EchoMatch: Partial-to-Partial Shape Matching via Correspondence Reflection

Supplementary Material

A. Implementation Details

EchoMatch. For both the overlap predictor and the feature extractor, we use Diffusion Net [56]. For the overlap predictor, we use three Diffusion Net blocks with a hidden dimension size of 16. Similar to [3], our feature extractor contains four Diffusion Net blocks with a hidden dimension size of 128. We train each learning-based method (DPFM and ours) for 60,000 iterations and use the final checkpoint for evaluation on CP2P24 and PSMAL. For BeCoS, we use the best-performing checkpoint on its validation split and report results on its test split. We use 50 eigenfunctions for our functional map module. For the predicted functional map, we impose structured regularization [47] with $\gamma = 0.5$ and $\lambda = 100$ as in [3]. Also, for the ground truth functional map, we impose regularization by treating eigenfunctions as coefficients. To this end, we choose $\gamma = 0.5$ and $\lambda = 10^6$. For our overlap prediction, we initialize the temperature parameter $\tau = 0.1$ for the soft point map and treat it as a learnable parameter during training. We establish the final point-to-point correspondences for every point in the predicted overlap region by computing the nearest neighbour of each vertex in \mathcal{F}_X in \mathcal{F}_Y . As described in the main paper, the total loss is the unweighted summation of three losses: functional map loss L_{fmap} , overlap loss L_{ov} and PointInfoNCE loss L_{nce}

$$L_{\text{total}} = L_{\text{ov}} + L_{\text{fmap}} + L_{\text{nce}}. \quad (7)$$

The PointInfoNCE loss contains two terms: a self-contrast term L_{self} and a cross-contrast term L_{cross}

$$L_{\text{nce}} = \lambda_{\text{self}} L_{\text{self}} + \lambda_{\text{cross}} L_{\text{cross}}. \quad (8)$$

We choose $\lambda_{\text{self}} = 0.1$ and $\lambda_{\text{cross}} = 0.1$.

Computational Resources. For the learning-based methods, DPFM and our method EchoMatch, we use five cores of an Intel Xeon Gold 6148 CPU with 36GB RAM and an NVIDIA RTX 5000 GPU with 32GB VRAM. For the axiomatic methods SM-COMB and GC-PPSM, we use an Intel Xeon E5-2697 with 16 cores and 36GB of RAM.

B. Dataset Split Details

We evaluate our method on three benchmark datasets for partial-to-partial shape matching: CP2P, PSMAL, and BeCoS. Below, we detail the dataset splits and configurations used in our experiments.

CP2P. The CP2P dataset (Cuts-Partial-to-Partial), first introduced in DPFM [3], is derived from the SHREC16 dataset [15] and the TOSCA dataset [10]. For our main experiments, we refer to the setup from [24] as CP2P24, where 120 shapes from the SHREC16 CUTS training set are used to generate 1164 training pairs, and evaluation is performed on 100 test pairs sampled from 153 shapes in the SHREC16 CUTS24 test set. For ablation studies, we refer to the original DPFM setup [3] as CP2P21, where the data is split into 242 training pairs and 61 validation pairs derived from SHREC16 CUTS training set.

PSMAL. The PSMAL dataset (PARTIALSMAL) [24], is derived from the SMAL [66] dataset and features non-isometric, normalised partial shapes of animals. The dataset includes 49 distinct shapes across 8 animal species. We follow a train/test split based on species, ensuring that the training and test sets contain different animal categories. This results in 273 training pairs and 100 test pairs.

BeCoS. BeCoS [23] is the most challenging dataset in our evaluation, containing non-isometric, realistically scaled partial shapes of humanoids and animals. It is the only partial-to-partial shape matching dataset with a structured train/validation/test split, containing 10,185 train, 137 validation, and 142 test instances. Evaluations are conducted in both directions for each pair. In all our experiments, we use a subset of the first 701 train instances, resulting in 1,402 training pairs. For ablation studies, we report results on the validation split.

C. Time Measurements

In Table A.1, we show the computation time for EchoMatch in comparison to the baseline methods. The axiomatic methods (SM-COMB and GC-PPSM) do not need any training time. Nevertheless, they are slower during inference. Both supervised learning-based methods (DPFM and EchoMatch) require similar time for training and inference.

D. Ablation Studies

D.1. Impact of Losses

We evaluate the influence of the different loss terms in Table A.2. To this end, we ablate on the BeCoS validation set [23] and CP2P21 dataset as described in Sec. B. For the evaluation, we consider both the overlap prediction in terms of the mean Intersection over Union (mIoU),

Method	Training (whole set)	Inference (per shape pair)
SM-COMB [54]	N/A	0.1h
GC-PPSM [24]	N/A	3.1h
DPFM [3]	4.8h	0.18s
Ours	5.4h	0.20s

Table A.1. We show the **mean computation time** for our method in comparison to the baseline methods. Axiomatic methods do not require any training time but are slow during inference. DPFM and our method require similar time for training and inference.

as well as the correspondence quality in terms of mean geodesic error (mGeoError). Our overlap loss L_{ov} is essential for achieving reasonable overlap predictions (without L_{ov} , mIoU is zero, cf. first row in Table A.2).

For the functional map loss L_{fmap} , we see a positive influence on the CP2P21 dataset due to its isometric nature and the low-pass smoothing effect from the functional map. On the more challenging BeCoS dataset, L_{fmap} deteriorates the performance, which might stem from the severe non-isometries and challenging high-frequency partialities present in the dataset.

The PointInfoNCE loss L_{nce} boosts the performance on the BeCoS dataset as it operates on high-frequencies and is more robust to non-isometries. Yet, L_{nce} alone without L_{fmap} is insufficient on the CP2P21 dataset showing inferior performance, which we attribute to the high-quality DINOv2 features on the CP2P21 dataset due to intra-class matchings. Specifically, in the BeCoS dataset, shapes from different classes (e.g. elephant and cat) are matched. Different geometric properties of shapes from different classes lead to less accurate DINOv2 features, which explains the greater importance of L_{nce} . On the other hand, the CP2P21 dataset only includes matches between the same shape category resulting in similar geometric properties and thus leading to more accurate DINOv2 features. This requires less contribution of the L_{nce} and the high-frequency information from L_{nce} in this context can potentially over emphasize fine-grained details, leading to overfitting.

We found that combining all losses is the best general trade-off. With that, we are able to use fixed relative loss weights and achieve overall high-quality performance without dataset-specific weight tuning.

D.2. Application to General Shape Matching

To evaluate the impact of the EchoModule we analyse the performance in terms of geodesic error in the partial-to-full (P2F) and in the full-to-full (F2F) case. For F2F, we apply our losses directly to echoed scores (we bypass the “Overlap DiffusionNet” since we have 100% overlap). Our module reduces geodesic error in both cases.

Losses	CP2P21		BeCoS	
	mIoU (\uparrow)	mGeoError (\downarrow)	mIoU (\uparrow)	mGeoError (\downarrow)
$L_{fmap} + L_{nce}$	0	6.43	0	8.03
$L_{ov} + L_{fmap}$	73.36	7.12	59.79	9.39
$L_{ov} + L_{nce}$	67.71	10.84	69.94	6.94
$L_{ov} + L_{fmap} + L_{nce}$	71.07	7.23	67.43	7.80

Table A.2. We show the mean Intersection over Union ($\times 100$) and mean geodesic error ($\times 100$) for **different loss combinations**. Overall, combining all three losses shows the best generalisation capabilities over both datasets.

Geo.Err.	w/o Echo	w/ Echo(Raw)	w/ Echo(Complete)
BeCoS P2F	5.53	5.30	5.24
BeCoS F2F	3.95	3.70	–

Table A.3. **Application to Partial-to-Full (P2F) and Full-to-Full (F2F) shape matching:** Our EchoModule reduces the geodesic error in both P2F and F2F settings.

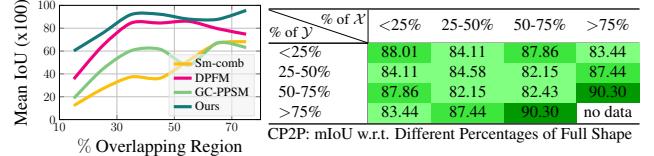


Figure A.1. **Overlap Percentage Analysis:** With increasing amount of overlap all method perform better in terms of mean IoU on the CP2P24 dataset (left). With high percentage of the full shape ($> 75\%$ and $50 - 75\%$) the mIoU improves (right).

E. Analysis of Pattern of Incompleteness

We analyse the overlap prediction (mIoU) of partial-to-partial shape matching methods on the CP2P24 dataset in terms of overlap percentage (see Figure A.1 left). All methods improve with larger overlaps. In the per-shape analysis (see Figure A.1 right) the mIoU increases when both shapes are more complete ($> 75\%$ and $50 - 75\%$).

F. Qualitative Results

F.1. More Qualitative Results on Different Datasets

We show additional qualitative results in Figure A.5 and A.6 on the CP2P24, PSMAL and BeCoS datasets. The axiomatic methods SM-COMB and GC-PPSM can only solve the matching on low resolution, which results in false patch-wise overlaps or false correspondences (see red ellipses in Figures A.5 and A.6). DPFM mostly struggles with inaccurate overlap predictions. Consequently, the correspondence predictions on the wrongly matched overlapping parts are often of low quality (see red ellipses in Figures A.5 and A.6).

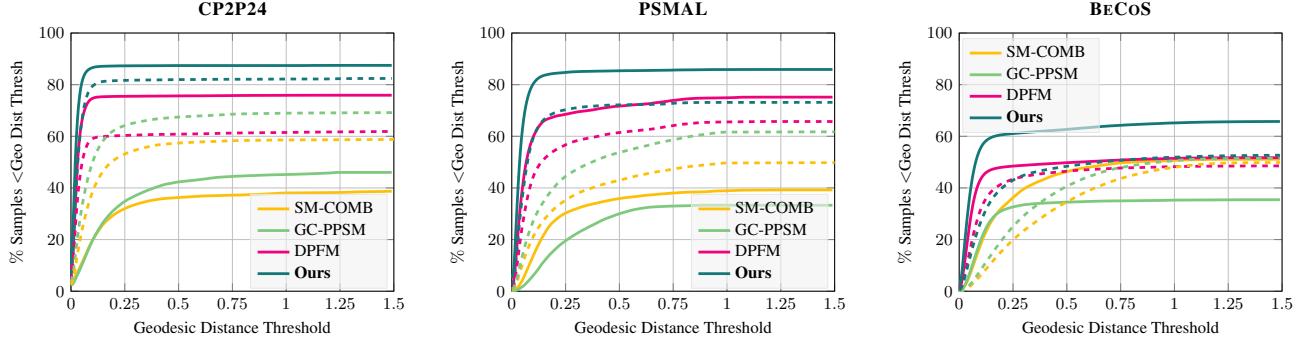


Figure A.2. We show the **uncut PCK curves** for the different partial-to-partial shape matching methods on three different datasets with DINOv2 features (solid). We also add the corresponding curves for spatial coordinate input (dashed) for completeness.

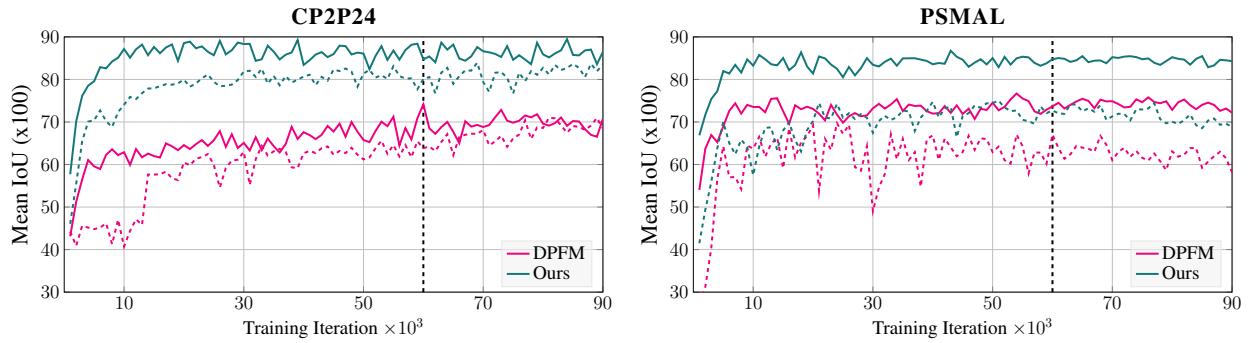


Figure A.3. **Training progress** comparison on CP2P24 and PSMAL datasets w.r.t the mIoU metric. We show the training curves for both DPFM and our method with DINOv2 features (solid) and spatial coordinates as inputs (dashed). Both methods are trained and stopped at 60 000 iterations. Our method consistently achieves higher mean IoU scores throughout the whole training.

F.2. Failure Cases

We show failure cases of our method in Figure A.4. Especially on the very challenging BeCoS dataset, our method shows inaccurate overlapping region predictions and inaccurate correspondence predictions. These likely stem from scanning artefacts which introduce significant noise.

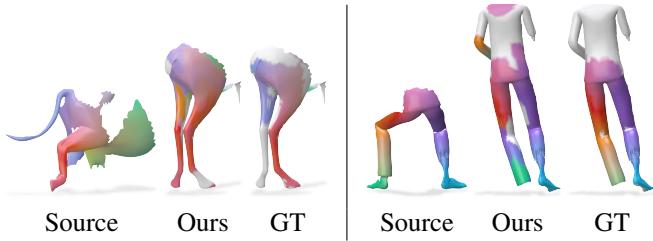


Figure A.4. We show **failure cases of our method** on the challenging BeCoS dataset. Due to the scanning artefacts the partial shapes have very challenging geometry, which makes it difficult for our method to predict good overlap regions.

G. Full Geodesic Error Curves

For better visualisation, we show cut geodesic error curves in Figure 5. For completeness, we add the uncut geodesic error curves in Figure A.2.

H. Training Details and Convergence Analysis

We provide a detailed analysis of the training progression for both our method and DPFM. Each model is trained and stopped at 60,000 iterations on the CP2P24 and PSMAL datasets. As shown in Figure A.3, our method exhibits faster convergence and achieves better performance across both datasets.

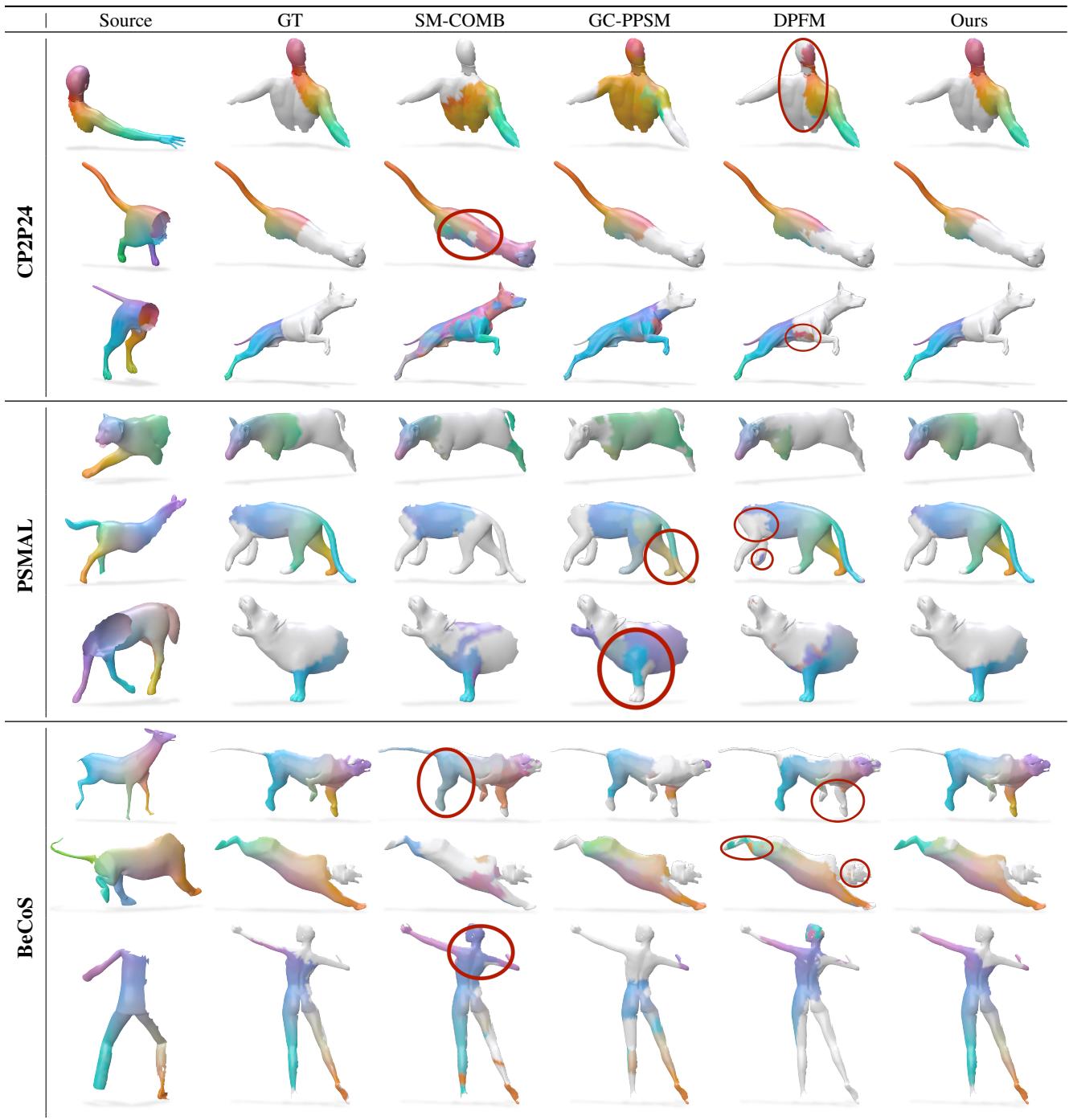


Figure A.5. We show **qualitative results on CP2P24, PSMAL and BeCoS** for all methods: the two axiomatic methods SM-COMB and GC-PPSM as well as the two learning based methods DPFM and our EchoMatch. Our method shows the most accurate overlap predictions and smoothest matchings. We mark incorrect predictions with red ellipses, which include both matching errors from GC-PPSM and SM-COMB (due to lower shape resolution) and overlap prediction errors from DPFM.

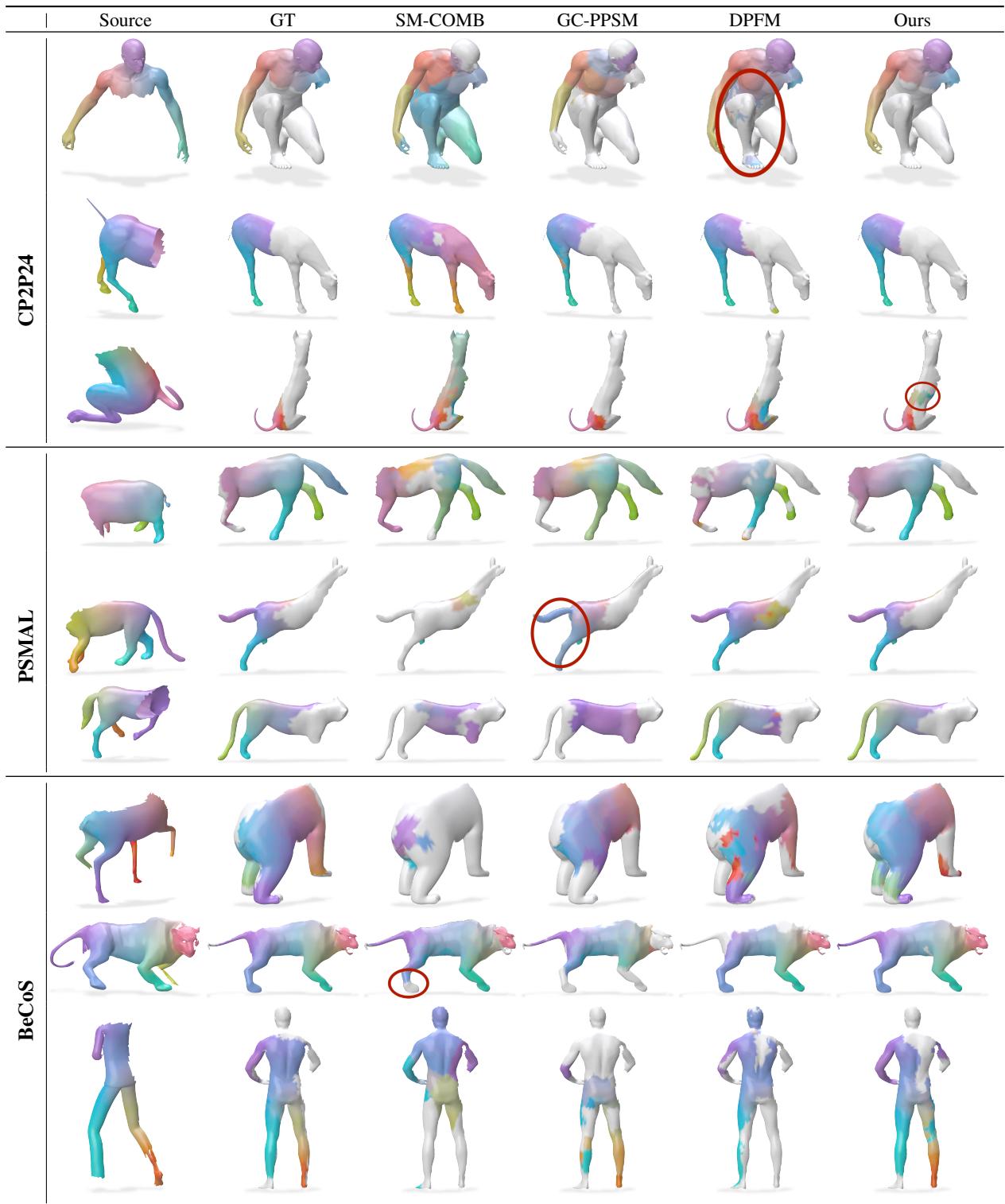


Figure A.6. **Additional qualitative comparison between EchoMatch and baseline methods.** While our method generally outperforms baselines as in Figure A.5, these examples also showcase some challenging cases where our method also fails.