

# 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  $L_{\text{fmap}}$ , overlap loss  $L_{\text{ov}}$  and PointInfoNCE loss  $L_{\text{ncc}}$

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

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

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

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<br>(whole set) | Inference<br>(per shape pair) |
|--------------|-------------------------|-------------------------------|
| SM-COMB [54] | N/A                     | 0.1h                          |
| GC-PPSM [24] | N/A                     | 3.1h                          |
| DPFM [3]     | 4.8h                    | 0.18s                         |
| <b>Ours</b>  | 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         | <b>5.24</b>       |
| BeCoS F2F | 3.95     | <b>3.70</b>  | –                 |

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.

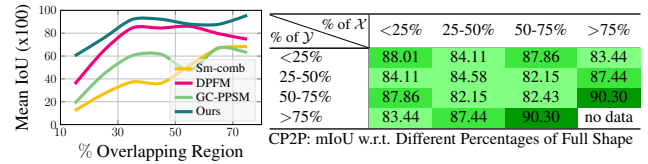


Figure A.1. **Overlap Percentage Analysis:** With increasing amount of overlap all methods perform better in terms of mean IoU on the CP2P24 dataset (left). With high percentage of the full shape ( $> 75\%$  and  $50 - 75\%$ ) the mIoU improves (right).

## 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).

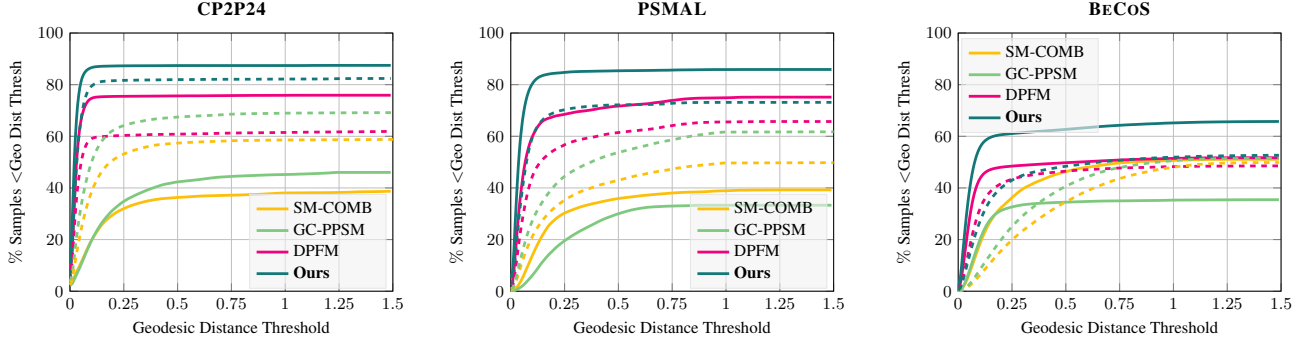


Figure A.2. We show the **uncut PCK curves** for the different partial-to-partial shape matching methods on three different datasets with DINOv2 features (solid). We also add the corresponding curves for spatial coordinate input (dashed) for completeness.

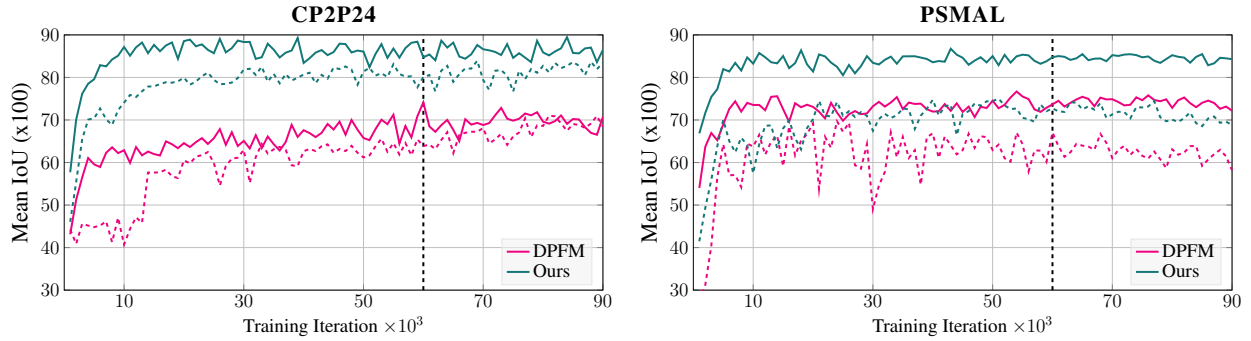


Figure A.3. **Training progress** comparison on CP2P24 and PSMAL datasets w.r.t the mIoU metric. We show the training curves for both DPFM and our method with DINOv2 features (solid) and spatial coordinates as inputs (dashed). Both methods are trained and stopped at 60 000 iterations. Our method consistently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

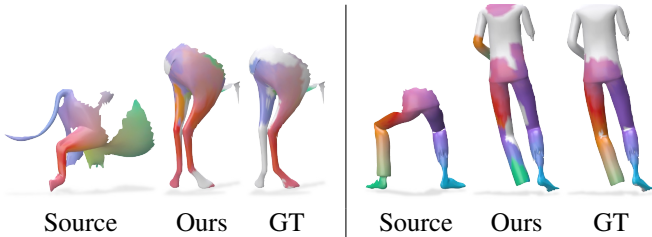


Figure A.4. We show **failure cases of our method** on the challenging BeCoS dataset. Due to the scanning artefacts the partial shapes have very challenging geometry, which makes it difficult for our method to predict good overlap regions.

## 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.




























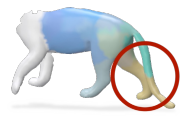





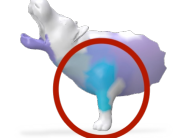




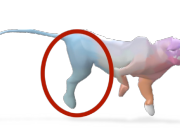















|        | Source  | GT  | SM-COMB   | GC-PPSM  | DPFM  | Ours  |
|--------|---|---|---|--|---|---|
| CP2P24 |    |    |    |    |    |    |
|        |    |    |    |    |    |    |
|        |    |    |    |    |    |    |
| PSMAL  |    |    |    |    |    |    |
|        |    |    |    |    |    |    |
|        |  |  |  |  |  |  |
| BeCoS  |  |  |  |  |  |  |
|        |  |  |  |  |  |  |
|        |  |  |  |  |  |  |

Figure A.5. We show **qualitative results on CP2P24, PSMAI 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.

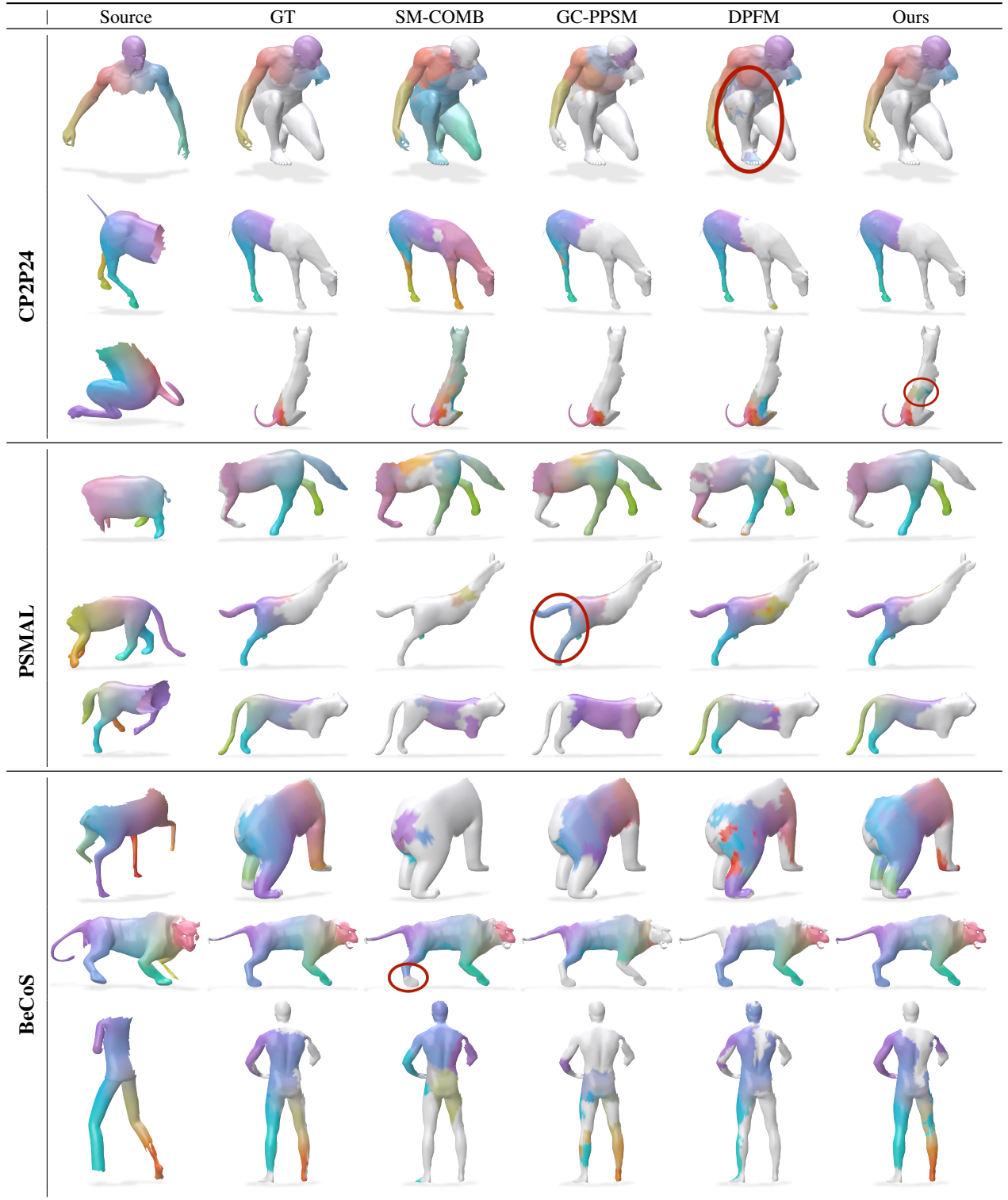


Figure A.6. **Additional qualitative comparison between EchoMatch and baseline methods.** While our method generally outperforms baselines as in Figure A.5, these examples also showcase some challenging cases where our method also fails.