

CoE: Deep Coupled Embedding for Non-Rigid Point Cloud Correspondences

Supplementary Material

In this supplementary material, we aim to provide additional implementation details, ablation studies and more experimental results. The document is organised as follows: first, we provide details about Laplace-Beltrami Operator in Sec. 8 and implementation details including necessary pre-processing steps in Sec. 9. In Sec. 10 we ablate our architecture and loss design and show an example how a challenging case is solved step-by-step by our careful design (Sec. 10.1). We further ablate the performance of our proposed method at different spectral resolutions, namely the dimension of our predicted embeddings (Sec. 10.2). Moreover, we analyse the performance of our method’s execution time with inputs of different sizes (Sec. 10.3). Then we elaborate the extension of our method to the partial setting in Sec. 11. Finally, we present more qualitative results on various datasets and failure cases in Sec. 12.

8. Laplace-Beltrami Operator

For a given function u defined on a Riemannian manifold \mathcal{M} , the Laplace-Beltrami Operator Δu measures how the function deviates from its average value within each local neighborhood, taking into account the geometry of \mathcal{M} . This property leads to its widespread application in computational geometry and computer graphics, especially when dealing with curved surfaces or manifolds. In practical applications, the discrete LBO approximates the continuous operator, enabling its use on graphs or discrete meshes.

A prevalent variant of the discrete LBO is the cotangent Laplacian, which is widely used due to its ability to approximate the LBO on discrete surfaces. It is defined based on the cotangent values of the angles around each vertex. The discrete formulation of the LBO at a single vertex i is represented as:

$$(\Delta u)_i \approx \frac{1}{2\mathcal{A}_i} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(u_i - u_j) \quad (9)$$

Here, \mathcal{A}_i denotes the area associated with vertex i , typically computed as one-third of the areas of the adjacent faces, and α_{ij} and β_{ij} are the angles opposite to the edge connecting vertices i and j . Based on the definition of the LBO: $\Delta = \mathbf{M}^{-1}\mathbf{L}$, the mass matrix \mathbf{M} and the stiffness matrix \mathbf{L} are defined as:

$$\mathbf{M} = \text{diag}(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_{n_S}) \quad (10)$$

$$\mathbf{L}_{ij} = \begin{cases} \omega_{ij} = -\frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}) & j \in \mathcal{N}(i) \\ -\sum_{j \in \mathcal{N}(i)} \omega_{ij} & j = i \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

| Geo. error ($\times 100$) | FAUST | SCAPE |
|---------------------------------------|------------|------------|
| Ablation study on loss terms | | |
| w/o L_{off}, L_o | 20.1 | 25.8 |
| w/o L_o | 11.2 | 15.6 |
| w/o L_c | 5.6 | 4.7 |
| Ablation study on network components | | |
| w/o ASAP | 3.9 | 4.3 |
| w/o Cross Attention | 4.0 | 3.4 |
| w/o ASAP, Cross Attention | 4.1 | 3.6 |
| Ablation study on shape communication | | |
| w/o Cross Attention, L_c | 4.6 | 4.9 |
| Ours | 3.7 | 3.2 |

Table 4. Ablation study of our loss and pipeline. Each loss term and network component contributes to reduce matching errors.

Finally, the eigenvalues Λ and eigenvectors Φ of the LBO Δ can be calculated using the following formula:

$$\mathbf{L}\Phi = \mathbf{M}\Lambda\Phi \quad (12)$$

9. Implementation Details

Our network is implemented in PyTorch [32]. The embedding extractor is based on the ASAP DiffusionNet with the default configuration published in [2] and the cross attention block is based on the implementation in [3]. We employ HKS [48] as descriptor functions $\mathbf{D}_{(\cdot)}$ in our pipeline, where we set its feature dimension $d = 512$. The dimension of the predicted (both intermediate and final) embeddings $\Psi_{(\cdot)}$ is set to be $k = 50$. Note that the same configuration is used across all our experiments to ensure a non-biased comparison. For more details, please refer to the supplementary material. The hyper-parameters in Eq. (7) are chosen as follows: $\mu_{\text{off}} = 1, \mu_o = 5e1$ and $\mu_c = 1e3$. Our network is trained using the Adam optimiser with a learning rate of $1e-3$ with a batch size of 1, except for SURREAL the batch size is set to 4.

As pre-processing, we approximately pre-align (using procrustes analysis or manually in blender) and normalise all shapes to the unit ball in each dataset, compute their point cloud LBOs [44] and eigen-decompose them to obtain the eigenvalues and eigenvectors, based on which the HKS is pre-computed as well.

10. Ablation Study

10.1. Architecture and Loss

In this section, we first explain the difference between DiffusionNet and our ASAP variant. Then we evaluate our proposed loss terms L_{off}, L_o and L_c . We discard L_{off} and L_o

together, and then L_o, L_c respectively, since L_{off} and L_o collectively encourages the desired basis structure. Next, we progressively disable the ASAP operation (by reverting to the original implementation of DiffusionNet) and the cross attention block to demonstrate the effectiveness of our proposed architecture design. Additionally, we evaluate the scenario of removing all cross-communication components: L_c and the cross attention block. All experiments follow the same configuration as in Sec. 5.2 and are evaluated on FAUST and SCAPE.

Compared to the original DiffusionNet, the ASAP version achieves smoother features by projecting the embedding $\hat{\Psi}'^{(i)}$ in layer i onto the Laplacian basis during the diffusion process in each layer of the diffusion block, and then projecting it back to the original space. This operation encourages the output embedding $\hat{\Psi}^{(i)}$ of each layer to approximate the properties of a smooth function, meaning that the embedding space can be better described using only low-frequency information. Specifically, this can be expressed as: $\hat{\Psi}^{(i)} = \Phi\Phi^\dagger\hat{\Psi}'^{(i)}$.

Results The quantitative results are reported in Tab. 4. Each loss term contributes to the accuracy of predicted correspondences. The L_c and the cross attention block facilitate communication cross shapes. The additional smoothness in the embedding induced by ASAP operation is also helpful. Fig. 7 illustrates that our design addresses the challenging case of crossed legs step-by-step.

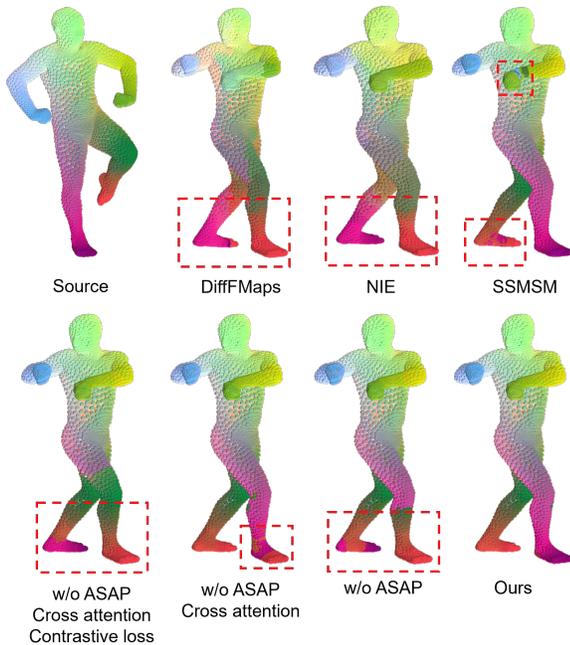


Figure 7. Visualisation of a challenging pair with crossed legs. We show our full design can successfully handle this challenge while all baseline methods fails (errors are highlighted in red).

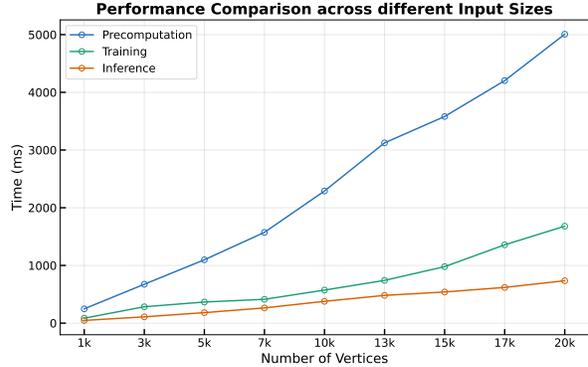


Figure 8. Our runtime vs. input point clouds size, ranging from 1k to 20k vertices. The runtime is measured per pair of shapes per forward pass, as our network processes two shapes simultaneously.

10.2. Dimension of Learned Embeddings

We analyze the sensitivity of the dimension k of our learned embeddings. In this experiment, we change k gradually from 10 to 80 and train in total 8 different networks on the SCAPE dataset. As shown in Fig. 10, with the increase of k , the mean geodesic error of our predicted dense correspondences decrease rapidly initially, then stagnates stably at a low level. This is expected since we design our off-diagonal loss such that the learned embeddings are “frequency-aligned”, enabling a relatively faithful representation of the shape already with a handful of embeddings.

10.3. Runtime Complexity

In this section, we analyse the execution time and scalability of our approach with respect to input size. Specifically, we divide the total execution time into three stages: Precomputation, Training, and Inference. We conduct this analysis using 20 shapes sampled from the FAUST dataset, measuring the average run-time for each stage on an NVIDIA A800 GPU and an Intel Xeon Gold 6348 CPU @ 2.60GHz.

Fig. 8 presents the run-time across different input sizes and stages. Although preprocessing involves some computational overhead, our method remains efficient and scalable, providing robust performance even for large and complex inputs. Furthermore, Fig. 9 demonstrates our approach’s capability to effectively match large-scale point clouds containing approximately 180k vertices.

11. Partial Shape Matching

As a proof-of-concept, we extend our method to the challenging task to match partial shapes. We take a full-partial pair (from the same shape category in SHREC16 [10]) during the training, and once the network is trained, we can directly match two partial shapes at inference time.

For this we introduce a new off-diagonal term for partial

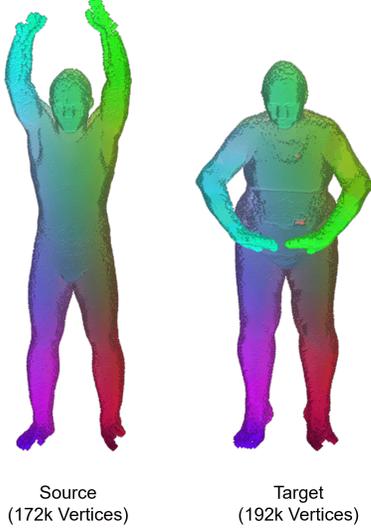


Figure 9. Matching results on MPI-FAUST [5] raw scan data. We downsample the original point cloud to 50k for better visualization. Our method can correctly handle point clouds of large sizes.

shapes analogous to the original one discussed in the main paper (cf. Eq. (4)).

$$L_{\text{off_partial}} = \|\Psi^T \mathbf{L} \Psi - \text{diag}(\Psi^T \mathbf{L} \Psi)\|_F \quad (13)$$

This adapted loss term only asks for the off-diagonal term of $\Psi^T \mathbf{L} \Psi$ to be as small as possible without any preference of frequency-alignment. This is reasonable because Eq. (4) will still be applied to the full shape and the frequency should be dictated by the full shape alone due to missing eigenvalues and eigenvectors under shape partiality [38]. Further we disable the orthogonal loss on the predicted embeddings of partial shapes. The reason behind is the embedding should be coupled to the one from the full shape, which contains frequencies that are absent in the partial shape. Fundamentally, it relates to deleting rows (or columns) of a Stiefel matrix will break the orthogonality, leading to a matrix which is not Stiefel anymore. The final loss reads as follows:

$$L_{\text{final}} = \mu_{\text{off_partial}} L_{\text{off_partial}} + \mu_{\text{off_full}} L_{\text{off_full}} + \mu_{\text{o_full}} L_{\text{o_full}} + \mu_c L_c. \quad (14)$$

We employ deep features extracted from SSMSM [7] (instead of HKS [48]). This choice was made because heat diffusion behaves differently under different partiality, resulting in even lower SNR of the final HKS, making it much harder to learn anything useful from it. Since our method is weakly supervised, we can exploit a test-time adaptation as in [8]. The hyperparameters are set as follows: $\mu_{\text{off_partial}} = \mu_{\text{off_full}} = 1$, $\mu_{\text{o_full}} = 5e3$ and $\mu_c = 5e3$.

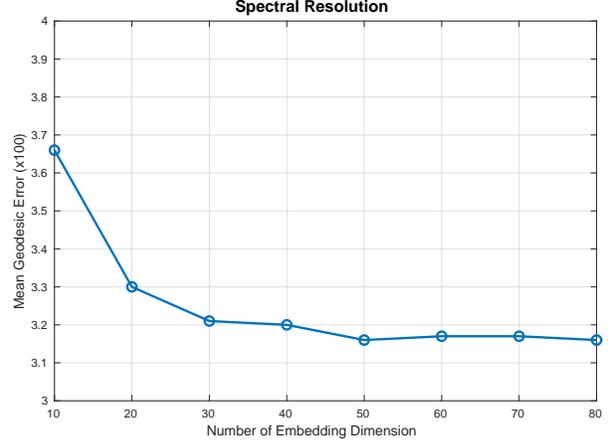


Figure 10. Illustration of mean geodesic error under different spectral resolutions. Our method is robust for different choice of spectral resolution.

We conduct experiments using the SHREC16 partiality dataset [10] and our proposed method show promising results especially in the HOLES sub-dataset (cf. Fig. 11). However we leave a comprehensive study as future work as partiality is challenging and deserves a thorough discuss itself.

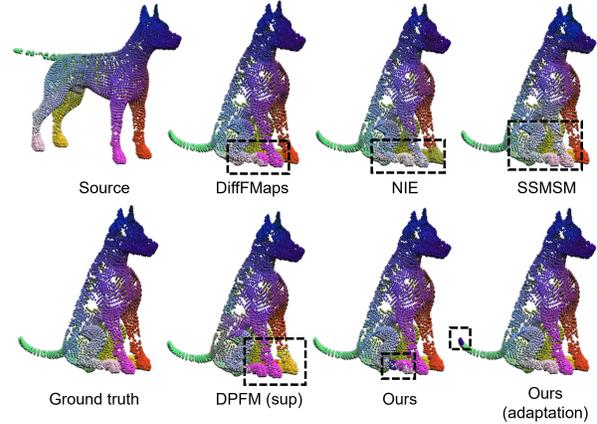


Figure 11. Qualitative result on SHREC16 HOLES. Ours performs reasonably well under this challenging setting.

12. More Qualitative Results

In this section, we present additional qualitative results including failure cases. See figure captions for explanation.

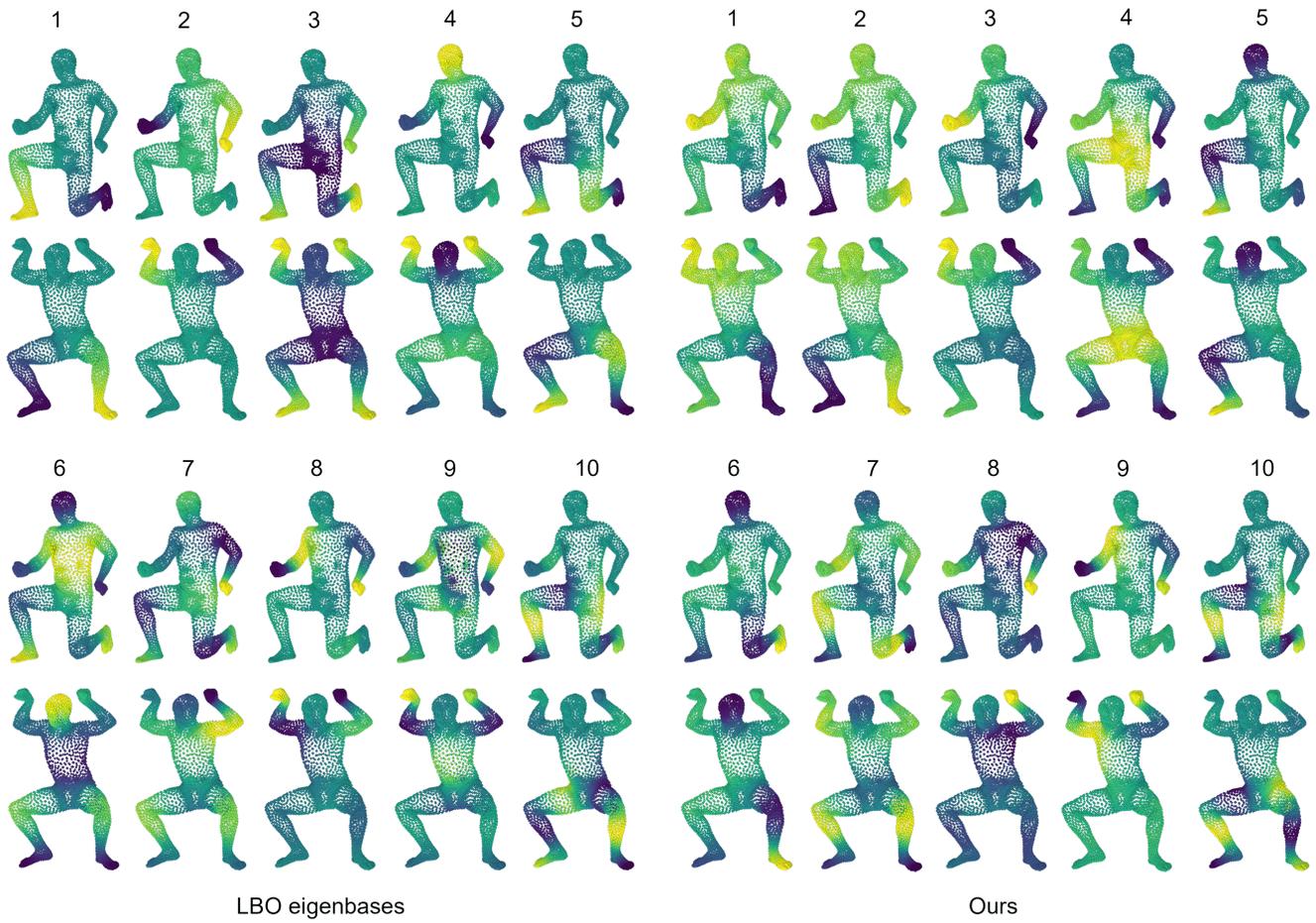


Figure 12. Visual comparison of our learned coupled embeddings vs. the LBO eigenbases. The first ten are shown. It can be seen that our learned embeddings are consistent while the LBO eigenbases suffer from sign flips and ambiguity in space corresponding to repeated eigenvalues.

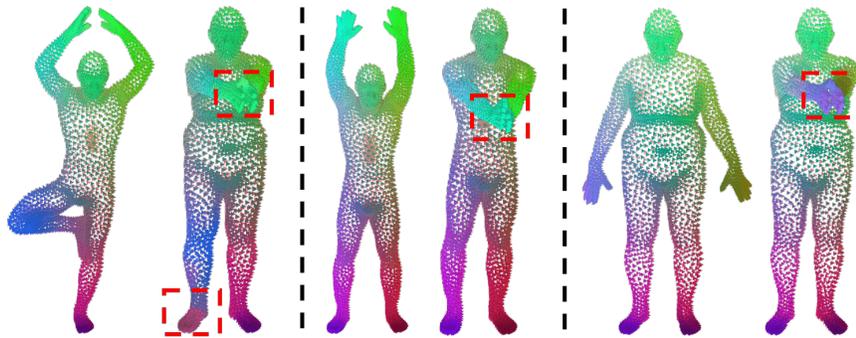


Figure 13. Failure cases on FAUST. All three failure examples relate to the touching hands, where the points of two hands are locally mixed and hard to separate. Note that this is the most challenging case for all point cloud methods in FAUST.



Figure 14. Qualitative results on SCAPE. Leftmost is the reference shape. Accurate correspondences are consistently obtained by our proposed method.



Figure 15. Qualitative results on TOPKIDS. Leftmost is the reference shape. Although our method can handle topological noise (*middle*), it still suffers from significant topological changes (*right*).

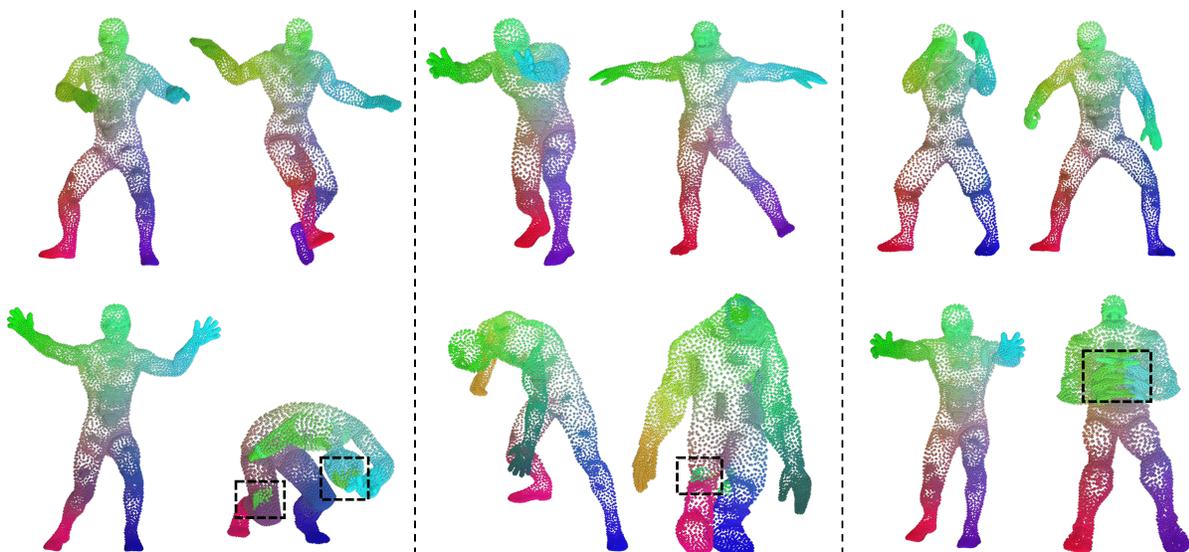


Figure 16. Qualitative results on DT4D-M. More qualitative non-isometric matching results (*top*). Failure cases mainly due to challenging topological noise (*bottom*).

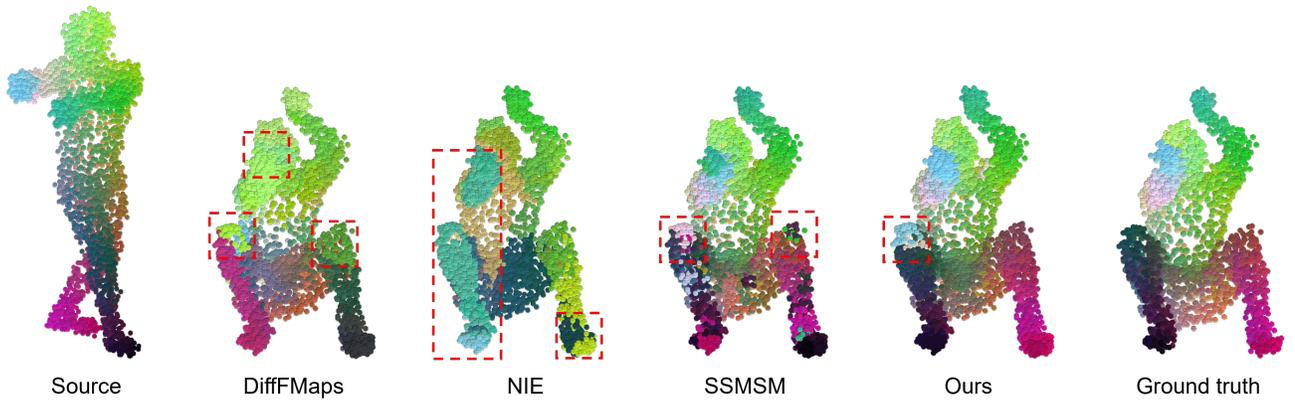


Figure 17. Robustness against additive noise. Ours produces stable correspondences under this noise compared to the baselines (errors highlighted in red).