Unsupervised Non-Rigid Point Cloud Matching through Large Vision Models

Anonymous Author(s)
Affiliation
Address
email

Figs/teaser.png

Figure 1: We match a challenging point cloud pair alien to the training set and visualize both maps and feature alignment. Our result surpasses that from purely LVM-based (w/o Geometry) and from purely geometry-based (w/o LVM) baselines in both ends. See text for more details.

Abstract

In this paper, we propose a novel learning-based framework for non-rigid point cloud matching, which can be trained *purely* on point clouds without any correspondence annotation but also be extended naturally to partial-to-full matching. Our key insight is to incorporate semantic features derived from large vision models (LVMs) to geometry-based shape feature learning. Our framework effectively leverages the structural information contained in the semantic features to address ambiguities arise from self-similarities among local geometries. Furthermore, our framework also enjoys the strong generalizability and robustness regarding partial observations of LVMs, leading to improvements in the regarding point cloud matching tasks. In order to achieve the above, we propose a pixel-to-point feature aggregation module, a local and global attention network as well as a geometrical similarity loss function. Experimental results show that our method achieves state-of-the-art results in matching non-rigid point clouds in both near-isometric and heterogeneous shape collection as well as more realistic partial and noisy data.

1 Introduction

Estimating dense correspondences between non-rigid 3D shapes is a fundamental task in computer vision and graphics, which plays a pivotal role in an array of applications including, including 3D reconstruction [63], 3D pose estimation [58], and animation [51] among others. In contrast to the significant progress [10, 56] on matching well-structured shapes (*i.e.,* triangular meshes), the advancement on *unstructured point clouds* is relatively lagged. Meanwhile, motivated by the prevalence and ease of point cloud data scanning in practice, we propose a novel *unsupervised* non-rigid shape matching framework, which is trained *purely* on point cloud base. Moreover, our framework can be naturally and effectively extended to partial point cloud matching, which fits squarely with the partial nature of point cloud scanning.

Before diving into our framework, we briefly overview the prior arts. Early approaches on unsupervised non-rigid point cloud matching [23, 34, 64, 17] leverage point cloud reconstruction as a proxy task to learn embeddings without correspondence labels. However, the reconstruction-based approaches attain training ease at the cost of intrinsic geometric understanding – the widely used Chamfer loss is purely *extrinsic*, falling short of efficiently capturing the geometric details of the underlying surface. To this end, a recent trend [42, 30, 11, 31] is to transplant the success of matching triangular meshes into the domain of point clouds. In essence, such approaches follow a self-supervised scheme built on the bijective mapping between a mesh and its vertex set. While achieving more competing performance than the above, these methods all require triangular meshes as training data, either for full or partial shape matching [11]. This constraint undoubtedly limits their piratical utility, especially in the tasks where high-quality meshes are non-trivial to achieve (***e.g.,*** 3D medical data).

Facing the aforementioned challenges, our key insight is to go *beyond* geometry and leverage a powerful tool from the world of a different dimension – large vision models (LVMs). First of all, LVMs are typically trained on datasets not only orders of magnitudes larger, but also significantly more versatile than the best possible in 3D domain. Based on such, LVMs have shown remarkable generalizability on understanding ever-changing objects in the wild, which would be highly desired in enhancing generalizability for our model. Secondly, LVMs are trained on images, which inherently are partial (in the sense of 3D world). The induced capacity of robustly encoding partiality is beyond valuable in dealing with partial point cloud matching.

In fact, there has been a recent trend on utilizing LVMs in 3D vision tasks, such as 3D model pre-training [65], shape segmentation [20], keypoint detection [60], and, most relevantly, 3D shape matching [1, 46]. Despite the simplicity and effectiveness, these methods essentially leverages LVMs as virtual annotator for generating auxiliary cues to assist 3D vision tasks. For instance, [46] uses DinoV2 to generate sparse landmarks as input to a second-stage neural surface mapping. Since DinoV2 (among many LVMs) is not designed for dense, fine-grained matching, such strategy can be sensitive to noisy LVM features. Moreover, most of the above are built on mesh render, which is non-trivial to extend to point clouds, especially those of partiality.

In light of the above, we propose for the first time a fine-grained end-to-end feature learning framework for unsupervised non-rigid point cloud matching, which make full use of both 3D geometric learning and pre-trained LVMs. Firstly, we propose an efficient module for aggregating pixel-to-point features, which adeptly assigns pixel-wise 2D representations to point-wise 3D point clouds. Secondly, we introduce a novel local and global attention network that refines the integration of visual and geometric features, transforming them into a more fine-grained canonical space. This attention network enhances the model's ability to capture details and complex relationships within the data. Finally, in addition to the conventional reconstruction loss, we propose a new geometry loss designed to further improve the model's performance by encouraging the preservation of geometric integrity.

Fig. 1 demonstrate a challenging task, in which we *directly* infer a pair of alien shapes from DT4D-H dataset, which are distinctive from SCAPE [5], the training set. LVM-dominated feature (w/o Geometry) leads to patch-wise correspondences, echoing the fact that it generally captures coarse semantics. On the other hand, geometry-dominated feature (w/o LVM) delivers smoother correspondences but fails to encode global structure, leading to severe mismatches around the hands. Finally, taking the best of both worlds, our learned feature leads to smooth and precise maps, surpassing the former two by a large margin. We also visualize the feature of the first channel on both shapes next to the target. It is obvious that our feature is by far more localized and cleaner than the counterparts.

We conduct a rich set of experiments to verify the effectiveness of our pipeline, highlighting that it achieves state-of-the-art results in matching non-rigid point clouds in both near-isometric and heterogeneous shape collections. Remarkably, it generalizes well despite the distinctiveness between the training set and test set. Moreover, our framework outperforms the competing methods in partial point clouds and noisy real scans.

2 Related Works

2.1 Non-rigid Shape Matching

Non-rigid shape matching is a long-standing problem in computer vision and graphics. Unlike the rigid counterpart, non-rigidly aligning shapes is more challenging owing to the complexity inherent in deformation models.

Originating from the foundational work on functional maps [50], along with a series of follow-ups [47, 27, 52, 44, 28, 39, 54, 10, 36, 19, 6, 56], spectral methods have made significant progress in addressing the non-rigid shape matching problem, yielding state-of-the-art performance. However, because of the heavy dependence of Laplace-Beltrami operators, DFM can suffer notable performance drop when applied to point clouds without adaptation [11]. In fact, inspired by the success of DFM, several approaches [30, 11, 31] have been proposed to leverage intrinsic geometry information carried by meshes in the training of feature extractors tailored for non-structural point clouds. When it comes to pure point cloud matching, there is a line of works[64, 34, 17] leverage point cloud reconstruction as the proxy task to learn embeddings without correspondence labels. Since intrinsic information is not explicitly formulated in these methods, they can suffer from significant intrinsic deformations and often generalize poorly to unseen shapes.

2.2 Large Vision Model for 3D Shape Analysis

Recently, Large Vision Models have become increasingly popular in due to their remarkable ability to understand data distributions from extensive image datasets. In the fields of shape analysis, [65] proposes an alternative to obtain superior 3D representations from 2D pre-trained models via Image-to-Point Masked Auto-encoders. [1] introduces a fully multi-stage method that exploits the exceptional reasoning capabilities of recent foundation models in language [48] and vision[35] to tackle difficult shape correspondence problems. In [46], before surface matching, the authors propose to use the features extracted from DINOv2 [49] of multi-view images of the shapes to perform co-alignment. In contrast to these approaches, which primarily utilize coarse patch features for sparse landmarks or semantic matching, our approach introduces an end-to-end method that aggregates pixel-level 2D features into point-wise 3D features.

2.3 Non-rigid Partial Shape Matching

While significant advancements have been made in full shape matching, there remains considerable room for improvement in estimating dense correspondences between shapes with partiality. Functional maps representation [53, 6, 11] has already been applied to partial shapes. However, both axiomatic and learning-based lines of work typically assume the input to be a *connected mesh*, with the exception of [11], which relies on graph Laplacian construction [55] in its preprocessing. For partial point cloud matching, axiomatic registration approaches [4, 62, 38] assume the deformation of interest can be approximated by local, small-to-moderate, rigid deformations, therefore suffer from large intrinsic deformations. Simultaneously, there's a growing trend towards integrating deep learning techniques [9, 8, 26, 37]. However, these methods often focus on addressing the partial sequence point cloud registration problem.

3 Methodology

Fig. 2 shows the overall pipeline. We first introduce our pixel-to-point feature aggregation method in Sec.3.1. Then our global and local attention network will be discussed in Sec.3.2. The training losses are described in Sec.3.3.

3.1 Pixel to Point Feature Aggregation

Depth aware projection: As shown in Fig2(b), given a point cloud P consisting of N points, we denote the $i-$th point by $p_i = (x_i, y_i, z_i)$. Following I2P-MAE [65], we project P on $xy-, yz-, xz-$plane to obtain three images. Without loss of generality, we consider the $xy-$plane and let (u_i, v_i) be the projected pixel of p_i, which is computed as follows:

$$u_i = \lfloor \frac{x_i - x_{\min}}{\Delta_{xy}} \times H \rfloor, v_i = \lfloor \frac{y_i - y_{\min}}{\Delta_{xy}} \times W \rfloor, \tag{1}$$

where $\Delta = \max\{x_{\max} - x_{\min}, y_{\max} - y_{\min}\}$ and $x_{\min}, x_{\max}$ are the minimum and maximum of the $x-$coordinates of P, and H, W are the pre-determined image resolution. On the other hand, Figs/pipeline.png

Figure 2: The schematic illustration of our pipeline.

since $z-$coordinates are eliminated, I2P-MAE proposes to assign sigmod(z_i) as the intensity of $(u_i, v_i), i = 1, 2, \cdots, N$, and 0 for all the unprojected pixels.

Despite of the simplicity, the above scheme eventually gray images with holes (as only projected pixels carry non-zero intensity), which are distinctive from the realistic training images used in LVMs. To alleviate the discrepancy, we propose to 1) apply a 3×3 mean filter on the gray images and 2) assign pseudo color on the pixel values with the `PiYG` colormap in MATLAB. We now denote by $I_{\hat{z}}, I_{\hat{x}}, I_{\hat{y}}$ to resulting images, where $\hat{x}$ indicates projection onto $xy-$ plane (and similarly for $\hat{x}, \hat{y}$).

2D to 3D feature aggregation: In I2PMAE [65], the obtained three projected images are fed into DINOv2 [49], resulting in a $D \times D \times C$ feature. Note that D is typically around 16, which is much smaller than H (or W). Thanks to the recent advance on *super-resolution* features from LVMs – FeatUp [21], we get rid of the resolution degradation and obtain

$$F^{img}_{\hat{z}} = \Theta(I_{\hat{z}}) \in \mathbb{R}^{H \times W \times C}, \tag{2}$$

where Θ is the per-pixel encoder of DinoV2-FeatUp [21] and C is the number of channels for each pixel. According to Eqn. 1, we obtain the point-wise feature of p_i via a simple pull-back:

$$f^i_{\hat{z}} = F_{\hat{z}}(u_i, v_i, :) \in \mathbb{R}^C. \tag{3}$$

We then have $F^{pt}_{\hat{z}} \in \mathbb{R}^{N \times C}$ by stacking $f^i_{\hat{z}}$ in order. We compute $F^{pt}_{\hat{x}}, F^{pt}_{\hat{y}}$ in the same manner. We emphasize that these computations are independent, as the pixel-point maps (Eqn. 1) vary. In the end, we arrive at

$$F^{pt}(P) = [F^{pt}_{\hat{z}}, F^{pt}_{\hat{x}}, F^{pt}_{\hat{z}}] \in \mathbb{R}^{N \times 3C}. \tag{4}$$

The above procedure returns a set of per-point features for the input P, which essentially carry the semantic information extracted by the LVM.

3.2 Local and Global Attention Network

In this part, we describe our Local and Global attention Network, which are depicted in Fig. 2(c).

Input feature: Given a point cloud P, we have computed the semantic per-point features based on LVMs (Eqn. 4). In order to exploit both semantic (image-based) and geometric (point-based) features, we propose to perform early fusion at the input stage as follows:

$$F^{in}(P) = \text{LBR}(F^{pt}(P)) + \gamma(P). \tag{5}$$

where $\gamma(P) \in \mathbb{R}^{N \times 384}$ is the positional encoding [45] and LBR is a module proposed in PCT [24] for non-linearly converting $F^{pt}(P)$ into the same dimension of $\gamma(P)$.

Architecture design: In non-rigid shape matching, both local and global features provide critical information for finding the precise correspondence. Intuitively, local features guide fine-detailed matching while suffering from global structure ambiguity due to self-similarities across the shape. Global features, on the other hand, provide structural descriptions for making full use of the local ones.

Motivated by the above, we propose a dual-pathway architecture in parallel, comprising global attention [61] and local attention [24] blocks. For each point, the global attention blocks systematically survey the features of the remaining points to achieve comprehensive global perceptual awareness.

On the other hand, the receptive field of the local attention blocks is constrained to the local neighborhood of a point. In particular, we highlight the key difference between the usage of local attention blocks in PCT [24] and ours: The former used a fixed neighborhood computed w.r.t the input spatial distribution, while we employ KNN search to connect with the nearest k features in the *latent space*. This design is inspired by DGCNN [59], and motivated by a typical challenge in non-rigid point cloud matching. Namely, points with small Euclidean distance are not necessarily close on the surface (*e.g.,* when a human put hand close to head).

The above dual-path design enables the extraction of profound information through a combination of layer-wise progression and cross-layer interactions. In the end, we leverage the fusion module, consisting of LBR and a three-layer stacked N2P [61] attention, to merge features from both global and local paths, resulting in our output feature. We refer readers to the appendix for more details.

3.3 Training Objectives and Matching Inference

In the following, we introduce the training losses, which consist of both reconstruction-based losses and our novel geometrical similarity loss. As shown in Fig. 2(a), our main model is a Siamese network. Given a pair of shapes (point clouds) $\mathcal{S}, \mathcal{T}$, $F_{\mathcal{S}}, F_{\mathcal{T}}$ are obtained by passing through the pixel-to-point feature aggregation (Sec. 3.1) and LG-Net (Sec. 3.2). We then estimate dense correspondences, $\Pi_{\mathcal{S}\mathcal{T}}$, based on the cosine similarity between features of each pair of points $x_i \in \mathcal{S}, y_j \in \mathcal{T}$, which is used to define the following losses.

Cross- and self-reconstruction losses are proposed in DPC [34], which aims to cross-construct the shape using latent proximity between source and target points and the coordinates. Specifically, the cross-construction process is computed as follows:

$$\hat{y}_{x_i} = \sum_{j \in \mathcal{N}_{\mathcal{T}}(x_i)} \frac{e^{s_{ij}}}{\sum_{l \in \mathcal{N}_{\mathcal{T}}(x_i)} e^{s_{il}}} y_j, \tag{6}$$

where $x_i \in \mathcal{S}, y_j \in \mathcal{T}$, and s_{ij} is the cosine similarity between the latent feature of them. $\mathcal{N}_{\mathcal{T}}$ represents the latent k-nearest neighbors of x_i in the target $F_{\mathcal{T}}$. The cross-construction of $\mathcal{T}$ by the source point cloud $\mathcal{S}$ is denoted $\hat{T}_S \in \mathbb{R}^{N \times 3}$, where $\hat{\mathcal{T}}^i_{\mathcal{S}} = \hat{y}_{x_i}$. The cross-construction is then be defined as:

$$\mathcal{L}_{cc} = \mathbf{CD}\left(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{T}}\right) + \mathbf{CD}\left(\mathcal{T}, \hat{\mathcal{T}}_{\mathcal{S}}\right), \tag{7}$$

where $\mathbf{CD}$ denotes Chamfer distance. In addition to the cross-construction Loss, we further employ a loss to enhance the smoothness within neighborhoods. This loss is equivalent to a special case of cross-construction, namely, the self-construction loss:

$$\mathcal{L}_{sc} = \mathbf{CD}\left(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{S}}\right) + \mathbf{CD}\left(\mathcal{T}, \hat{\mathcal{T}}_{\mathcal{T}}\right). \tag{8}$$

Mapping loss is also proposed in DPC [34], which enforces the mapped points of neighboring points in $\mathcal{S}$ to be close to each other as well in $\mathcal{T}$. Specifically, it is defined as

$$\mathcal{L}_m = \mathcal{L}_m^{(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{T}})} + \mathcal{L}_m^{(\mathcal{T}, \hat{\mathcal{T}}_{\mathcal{S}})}, \tag{9}$$

where $\mathcal{L}_m^{(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{T}})}$ denotes the mapping loss from $\mathcal{S}$ to $\mathcal{T}$

$$\mathcal{L}_m^{(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{T}})} = \frac{1}{N k_m} \sum_i \sum_{l \in N_{\mathcal{S}}(x_i)} e^{-\|x_i - x_l\|_2^2 / \alpha} \left\| \hat{y}_{x_i} - \hat{y}_{x_l} \right\|_2^2, \tag{10}$$

where k_m, α are fixed constants. We then define similarly on the opposite direction $\mathcal{L}_m^{(\mathcal{T}, \hat{\mathcal{T}}_{\mathcal{S}})}$.

Geometrical similarity: Previous methods [34, 17, 25] mainly rely on the above losses. It is worth noting, though, the involved cosine similarity emphasizes more on the *angular* difference between features, which falls short of constraining features from the perspective of *magnitude*. Inspired by such, we propose geometrical similarity loss for taking magnitude into consideration.

In particular, we notice NIE [30] leverages geodesic supervision to enforce the Euclidean distance between learned feature to ensemble geodesic. However, estimating accurate geodesics on point clouds is a non-trivial but also heavy task. We therefore adopt the heat method [16] for point clouds to compute $\mathbf{M}_{\mathcal{S}}$ as the approximated geodesic distance matrix of $\mathcal{S}$.

On the other hand, for each learned feature of $x_i \in \mathcal{S}$, $F_{\mathcal{S}}^i$, we consider $\mathrm{NN}(i) = \{j_1, j_2, \cdots, j_k\}$ be the set of ordered indices of the nearest neighborhood in the latent space and compute the distance vector $d_{\mathcal{S}}^i = [\|F_{\mathcal{S}}^i - F_{\mathcal{S}}^{j_1}\|, \|F_{\mathcal{S}}^i - F_{\mathcal{S}}^{j_1}\|, \cdots, \|F_{\mathcal{S}}^i - F_{\mathcal{S}}^{j_k}\|]$. Similarly, we can construct another vector given $\mathbf{M}_{\mathcal{S}}$ and $\mathrm{NN}(i)$, *i.e.,* $m_{\mathcal{S}}^i = [\mathbf{M}_{\mathcal{S}}(i, j_1), \mathbf{M}_{\mathcal{S}}(i, j_2), \cdots, \mathbf{M}_{\mathcal{S}}(i, j_k)]$.

While it seems natural to minimize the residual between the above two vectors for each point, we opt for the following loss given the potential noise in estimating geodesics on unstructured point clouds:

$$\mathcal{L}_{geo}^{\mathcal{S}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \frac{d_{\mathcal{S}}^i \cdot m_{\mathcal{S}}^i}{\|d_{\mathcal{S}}^i\| \|m_{\mathcal{S}}^i\|}). \tag{11}$$

Similarly, we define the geometrical similarity loss for $\mathcal{T}$:$\mathcal{L}_{geo} = \mathcal{L}_{geo}^{\mathcal{S}} + \mathcal{L}_{geo}^{\mathcal{T}}$.

The overall objective function of our point correspondence learning scheme is:

$$\mathcal{L}_{total} = \lambda_{cc}\mathcal{L}_{cc} + \lambda_{sc}\mathcal{L}_{sc} + \lambda_m\mathcal{L}_m + \lambda_{geo}\mathcal{L}_{geo}, \tag{12}$$

where $\lambda_{cc}, \lambda_{sc}, \lambda_m, \lambda_{geo}$ are hyper-parameters, balancing the contribution of the different loss terms.

Partial matching loss: In the above, we entail the losses for training full-to-full non-rigid point cloud matching. Remarkably, our formulation can be easily extended to the challenging scenario of partial-to-full matching. In fact, we simply modify $\mathcal{L}_{cc}$ to a unilateral loss, i.e. only $\mathbf{CD}\left(\mathcal{S}, \hat{\mathcal{S}}_{\mathcal{T}}\right)$ is considered, and set $\lambda_m = 0$.

Inference: At inference time, we choose the nearest latent cross-neighborhood of $x_i \in \mathcal{S}$ to be its corresponding point by KNN [15], thus get the shape matching result between point cloud $\mathcal{S}$ and $\mathcal{T}$.

4 Experiments

Dataset: We evaluate our method with several state-of-the-art techniques for estimating correspondences on a set of benchmarks as follows. **SCAPE_r:** The remeshed version of the SCAPE dataset[5] comprises 71 human shapes. We split the first 51 shapes for training and the rest 20 shapes for testing; **FAUST_r:** The remeshed version of FAUST dataset [7] comprises 100 human shapes. We split the first 80 shapes for training and the rest 20 for testing. **SHREC'19_r:** The remehsed version of SHREC19 dataset[43] comprises 44 shapes. We pair them into 430 annotated examples provided by [43] for testing. **DT4D-H:** A dataset from [41] comprises 10 categories of heterogeneous humanoid shapes. Following [31], we use it solely in testing, and evaluating the inter-class maps split in [41]. **SHREC'07-H:** A subset of SHREC'07 dataset [22] comprises 20 heterogeneous human shapes. We use it solely in testing. **TOSCA:** Dataset from [66] comprises 41 different shapes of various animal species. Following [34, 17], we pair these shapes to create both for training and evaluation, respectively. **SHREC'16:** Partial shape dataset SHREC'16 [14] includes two subsets, namely CUTS with 120 pairs and HOLES with 80 pairs. Following [6, 11], we train our method for each subset individually and evaluate it on the corresponding unseen test set (200 shapes for each subset). Moreover, we further conduct some practical experiments on partial real scan dataset processed from [33] and medical dataset from [3].

Baseline: We compare our method with a set of competitive baselines, including methods that can both train and test on point cloud; methods required mesh for geometry-based training but inference with point cloud. Methods are labelled [U], [S], [W] as unsupervised, supervised, weakly supervised.

Evaluation metric: Though we focus on the matching of point clouds, we primarily employ the widely-accepted geodesic error normalized by the square root of the total area of the mesh, to evaluate the performance of all methods.

Hyper-parameters: In Equation 12, $\lambda_{cc}, \lambda_{sc}, \lambda_m, \lambda_{geo}$ are normally set to 1, 10, 1, and 0.5 respectively. Model training utilizes the AdamW [40] optimizer with $\beta = (0.9, 0.99)$, learning rate of 2e-3, and batch size of 6. We provide more details of hyper-parameters in Tab. 9 in the appendix.

Table 1: Quantitative results on SCAPE_r, FAUST_r and SHREC'19_r in terms of mean geodesic errors ($\times 100$). The **best** results from the pure point cloud methods in each column are highlighted.

Method	Train / Test	SCAPE_r			FAUST_r		
		SCAPE_r	FAUST_r	SHREC'19_r	FAUST_r	SCAPE_r	SHREC'19_r
3D-CODED[S] [23]	Trained on Mesh	31.0	33.0	\	2.5	31.0	\
TransMatch[S] [57]		18.6	18.3	38.8	2.7	33.6	21.0
DiffFMaps[S] [42]		12.0	12.0	17.6	3.6	19.0	16.4
NIE[W] [30]		11.0	8.7	15.6	5.5	15.0	15.1
SSMSM[W] [11]		4.1	8.5	7.3	2.4	11.0	9.0
CorrNet3D[U] [64]	Trained on PCD	58.0	63.0	\	63.0	58.0	\
SyNoRiM[S] [26]		9.5	24.6	\	7.9	21.9	\
DPC[U] [34]		17.3	11.2	28.7	11.1	17.5	31.0
SE-ORNet[U] [17]		24.6	22.8	23.6	20.3	18.9	23.0
Ours-w/o LVM[U]		13.8	10.4	14.7	8.5	16.1	15.8
Ours[U]		**7.6**	**7.3**	**9.5**	**4.6**	**12.4**	**11.5**

Table 2: Quantitative results on DT4D-H and SHREC'07-H in terms of mean geodesic errors ($\times 100$). The **best** results from the pure point cloud methods in each column are highlighted.

Method	Train / Test	SCAPE_r		FAUST_r	
		DT4D-H	SHREC'07-H	DT4D-H	SHREC'07-H
TransMatch[S] [57]	Trained on Mesh	25.3	31.2	26.7	25.3
DiffFMaps[S] [42]		15.9	15.4	18.5	16.8
NIE[W] [30]		12.1	13.4	13.3	15.3
SSMSM[W] [11]		8.0	37.7	11.8	42.2
DPC[U] [34]	Trained on PCD	21.7	17.1	13.8	18.1
SE-ORNet[U] [17]		15.5	27.7	12.2	20.9
Ours-w/o LVM[U]		10.3	14.3	11.2	16.4
Ours[U]		**7.7**	**8.9**	**9.0**	**8.9**

4.1 Experimental Results

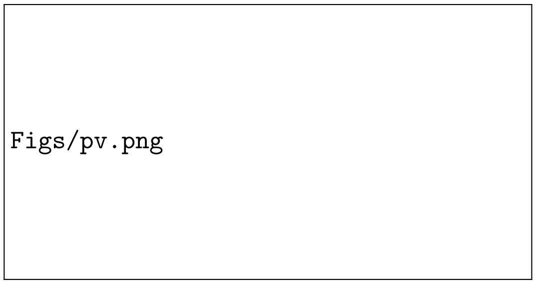
Near-isometric benchmarks: As illustrated in Tab. 1, our method consistently outperforms other pure point cloud methods in all settings. Especially, our method achieves a promising performance improvement of over **50%** compared to the previous SOTA approaches (**9.5** vs. 23.6; **11.5** vs. 23.0) in SCAPE_r/SHREC19'_r case and FAUST_r/SHREC19'_r case, i.e., many previous methods performs well in the standard seen datasets but generalizes poorly to unseen shapes. Remarkably, our method also indeed outperforms all of the baselines even the mesh-required method like SSMSM [11] (**7.3** vs. 11.2) in SCAPE_r/FAUST_r case.

Non-isometric benchmarks: We stress test our method on challenging non-isometric datasets including SHREC'07-H and DT4D-H. Our method achieves the SOTA performance for all kinds of methods as shown in Tab. 2, which indicates the excellent generalization ability to some unseen challenging cases. 1) Regarding DT4D-H, We follow the test setting of AttentiveFMaps [36], which only considers the more challenging inter-class mapping for testing of DT4D-H. The test on this non-isometric benchmark further confirms the robustness of our approach. 2) SHREC'07-H dataset comprises 20 heterogeneous human shapes with vertex numbers ranging from 3000 to 15000 and includes topological noise. Our method achieves a performance improvement of over **30%** compared to the previous SOTA approaches (**8.9** vs. 13.4; **8.9** vs. 15.3).

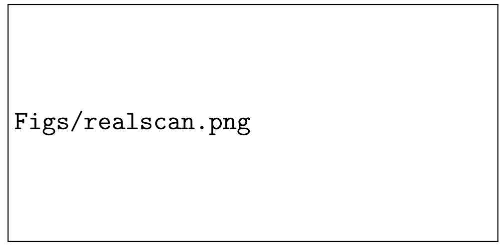
We attribute the above success to that the per-point features aggregated from LVMs carry rich semantic information, help to identify correspondence at the coarse level. In addition to that, our final features are further boosted by the geometric losses, leading to strong performance.

Partial matching benchmarks: As shown in Sec. 3.3, our framework can be easily adapted for unsupervised partial-to-full shape matching. We evaluate our method in two types of partial shape matching, including the challenging SHREC'16[14] Cuts and Holes benchmark and two partial-view benchmarks built on SCAPE_r and FAUST_r datasets by ourselves, where we employ raycasting from the center of each face of a regular dodecahedron to observe the shapes, resulting 12 partial view point clouds. In all cases, we match a partial point cloud with a given null (complete) point cloud.

The challenge of partial view matching arises from the presence of numerous disconnected components in the partial shapes, and the sampling of partial point clouds differs from that of complete shapes. We split the training and testing set consistent with those of SCAPE_r, FAUST_r, respectively. As illustrated in Tab. 3, our method outperforms the recent unsupervised method SSMSM [11] in 3 out of 4 test cases, which requires meshes for training. Fig. 3(a) further shows qualitative that our framework outperforms the competing methods including DPFM [6], which is based on mesh input as well.



(a) Partial view of SCPAE



(b) Noisy partial real scans

Figure 3: Qualitative results of SCAPE-PV and noisy real scans.

Table 3: Quantitative results on partial cases including SCAPE-PV, FAUSR-PV and SHREC'16 in terms of mean geodesic errors ($\times 100$). * indicates its original checkpoint using SURREAL190K. The **best** results from the pure point cloud methods in each column are highlighted.

Method	Train/Test	SCAPE-PV		SHREC'16-CUTS	SHREC'16-HOLES
		SCAPE-PV	FAUST-PV	CUTS	HOLES
ConsistFMaps unsup[U] [10]	Trained on Mesh	/	/	26.6	27.0
DPFM unsup[U] [6]		11.5	15.2	20.9	22.8
HCLV2S*[S] [29]		8.7	5.3	/	/
SSMSM[W] [11]		8.8	8.0	12.2	16.7
DPC[U] [34]	Trained on PCD	13.6	14.5	32.9	32.5
SE-ORNet[U] [17]		15.4	13.9	40.5	27.6
Ours-w/o LVM[U]		10.1	9.2	36.4	29.3
Ours[U]		**8.4**	**7.6**	**21.2**	**15.3**

Table 4: Generalization performance of the checkpoint trained on sampled point cloud with fixed 1024 points of SHREC'19. We test this checkpoint on the more dense original point cloud. The **best** is highlighted.

Method	DPC[U] [34]	SE-ORNet[U] [17]	Ours[U]
SHREC'19 (1024)	5.6	5.1	**4.72**
SHREC'19 (Ori.)	6.1 (+8.93%)	5.9 (+15.69%)	**4.73 (+0.21%)**

Regarding purely point cloud-based baselines, we modify the shared loss of [34, 17] to adapt partiality in the same way. In the end, we achieve the SOTA compared to them, exhibiting a significant over **45%** superiority in partial view matching (**8.4** vs. 13.6; **7.6** vs. 13.9) and over **35%** superiority in cuts/holes setting (**21.2** vs. 32.9; **15.3** vs. 27.6).

Generalization & robustness analysis: Reconstruction-based methods [34, 17, 25] typically perform down-sampling to $n = 1024$ points for *both* training and testing. On the one hand, over down-sampling leads to loss of geometric details (*e.g.,* human fingers). On the other hand, point clouds scanned in reality typically consist of tens of thousands of points, which is much denser. Such resolution gap can pose great challenge for purely geometric methods. To see that, we perform generalization test on the SHREC'19 benchmark. More specifically, we use the checkpoint trained on down-sampled data released by the regarding authors to evaluate performance in both down-sampled test data (1024 points) and original test data (~ 5000 points). As shown in Tab. 4, DPC and SE-ORNet [34, 17] both experience a degradation more than **8%**. On the other hand, our method only yields a 0.21% drop and achieves the best performance in both cases. Beyond the quantitative results, we also report qualitatively the generalization performance on TOSCA benchmark following the same setting as above, see Fig. 4 for more details.

We attribute the above robustness to our introduction of LVMs in point feature learning, which effectively compensates for the discrepancy of low-resolution geometry. Last but not least, we also compare our method following the same training scheme and evaluation protocol as [17]. The quantitative results are reported in Tab. 7 in the appendix, which again conforms our superiority over the baselines.

4.2 Realworld Applications

In this part, we showcase the utility of our framework in two real-world applications: **Matching real scans:** The Panoptic dataset [32] consists of partial point clouds derived from multi-view RGB-D images. We randomly select a subset of these views to recover partial point clouds. As shown in Fig. 3(b), we transfer texture from the source shape (left-most of each row) to target via maps from ours, DPC [34], SE-ORNet [17]. Our method demonstrates smoother texture transfer compared to

Figs/tosca.png

Figure 4: Qualitative results of TOSCA. Our method demonstrates enhanced generalization capabilities when transitioning from sparse point clouds in training to dense point clouds in testing.

Table 5: Statistical shape analysis on spleen medical dataset in terms of chamfer distance. The **best** is highlighted.

Model	PN-AE [2]	DG-AE [59]	CPAE [13]	ISR [12]	DPC [34]	Point2SSM [3]	Ours
CD (mm)	43.7	43.5	61.3	17.6	10.6	3.4	**2.9**

Table 6: Mean geodesic errors ($\times 100$) on different ablated settings, the models are all trained on SCAPE_r and test on SCAPE_r.

w/o pointwise proj	w/o LG-NET	w/o Geo. Loss	w/o Featup	w/o PE	w/o LA-NET	w/o GA-NET	w/o Fusion	Full
33.1	19.6	9.1	9.0	8.6	9.8	9.8	9.7	7.6

baselines (see particularly facial details and strips in the T-shirt); **Statistical shape models(SSM) for medical data:** Following Point2SSM [3], we test our method on the anatomical SSM tasks. We stick to the regarding experimental setting and report our score in the spleen subset. As shown in Tab. 5, our method outperforms the second best by a 14% relative error reduction.

4.3 Ablation Study

We first justify our overall design in Tab. 6, where we sequentially remove each building block from our pipeline and train/test model on SCAPE_r. The performance gaps well support our claims. Beyond that, in Tab. 1, Tab. 2 and Tab. 3, we report experimental results [w/o LVM] to demonstrate that the coarse semantic representations extracted by LVMs play a crucial role throughout the pipeline, whether it is in the full or partial settings. Finally, we highlight that we have also performed robustness evaluation regarding noisy data and rotation perturbations in Sec. B.3.

5 Conclusion and Limitation

In this paper, we address the challenge of unsupervised non-rigid point cloud matching. In conclusion, we proposed a novel learning-based framework for non-rigid point cloud matching that can be trained purely on point clouds without correspondence annotations and extends naturally to partial-to-full matching. By incorporating semantic features from large vision models (LVMs) into geometry-based shape feature learning, our framework resolves ambiguities from self-similarities and demonstrates strong generalizability and robustness. Our method achieves state-of-the-art results in matching non-rigid point clouds, even in challenging scenarios with partial and noisy data.

Limitation & Future Work The primary limitation of our method is its assumption of roughly aligned input point clouds. In the future, we plan to further explore the ability of LVM to address this limitation.

References

[1] Ahmed Abdelreheem, Abdelrahman Eldesokey, Maks Ovsjanikov, and Peter Wonka. Zero-shot 3d shape correspondence. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–11, 2023.

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. Learning representations and generative models for 3d point clouds. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[3] Jadie Adams and Shireen Elhabian. Point2ssm: Learning morphological variations of anatomies from point cloud. *arXiv preprint arXiv:2305.14486*, 2023.

[4] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. 2007.

[5] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape Completion and Animation of People. 2005.

[6] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021.

[7] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3794–3801, 2014.

[8] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. In *NeurIPS*, volume 33, pages 18727–18737, 2020.

[9] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *CVPR*, pages 7002–7012, 2020.

[10] Dongliang Cao and Florian Bernard. Unsupervised deep multi-shape matching. In *ECCV*, 2022.

[11] Dongliang Cao and Florian Bernard. Self-supervised learning for multimodal non-rigid shape matching. In *CVPR*, 2023.

[12] Nenglun Chen, Lingjie Liu, Zhiming Cui, Runnan Chen, Duygu Ceylan, Changhe Tu, and Wenping Wang. Unsupervised learning of intrinsic structural representation points. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9121–9130, 2020.

[13] An-Chieh Cheng, Xueting Li, Min Sun, Ming-Hsuan Yang, and Sifei Liu. Learning 3d dense correspondence via canonical point autoencoder. *Advances in Neural Information Processing Systems*, 34:6608–6620, 2021.

[14] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, Y Sahillioğlu, et al. Shrec'16: Partial matching of deformable shapes. In *Eurographics Workshop on 3D Object Retrieval, EG 3DOR*, pages 61–67. Eurographics Association, 2016.

[15] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[16] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, Oct. 2017.

[17] Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. Se-ornet: Self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5364–5373, 2023.

[18] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019.

[19] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *CVPR*, pages 742–751, 2022.

[20] Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. *arXiv preprint arXiv:2311.17024*, 2023.

[21] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track. *SHREC competition*, 8(7):7, 2007.

[23] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *ECCV*, 2018.

[24] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021.

[25] Jianfeng He, Jiacheng Deng, Tianzhu Zhang, Zhe Zhang, and Yongdong Zhang. Hierarchical shape-consistent transformer for unsupervised point cloud shape correspondence. *IEEE Transactions on Image Processing*, 2023.

[26] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J. Guibas, and Shi-Min Hu. Multiway Non-rigid Point Cloud Registration via Learned Functional Map Synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[27] Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. In *Computer Graphics Forum*, volume 36, pages 151–163. Wiley Online Library, 2017.

[28] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020.

[29] Xiangru Huang, Haitao Yang, Etienne Vouga, and Qixing Huang. Dense correspondences between human bodies via learning transformation synchronization on graphs. In *NeurIPS*, 2020.

[30] Puhua Jiang, Mingze Sun, and Ruqi Huang. Neural intrinsic embedding for non-rigid point matching. In *CVPR*, 2023.

[31] Puhua Jiang, Mingze Sun, and Ruqi Huang. Non-rigid shape registration via deep functional maps prior. In *NeurIPS*, 2023.

[32] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[33] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[34] Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *2021 International Conference on 3D Vision (3DV)*, pages 1442–1451. IEEE, 2021.

[35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[36] Lei Li, Nicolas Donati, and Maks Ovsjanikov. Learning multi-resolution functional maps with spectral attention for robust shape matching. In *NeurIPS*, 2022.

[37] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *CVPR*, 2022.

[38] Yang Li and Tatsuya Harada. Non-rigid point cloud registration with neural deformation pyramid. *Advances in Neural Information Processing Systems*, 35:27757–27768, 2022.

[39] Or Litany, Tal Remez, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, 2017.

[40] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[41] Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. Smooth non-rigid shape matching via effective dirichlet energy optimization. In *2022 International Conference on 3D Vision (3DV)*, pages 495–504. IEEE, 2022.

[42] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020.

[43] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, et al. Shrec'19: matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019.

[44] Simone Melzi, Jing Ren, Emanuele Rodolà, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *Proc. SIGGRAPH Asia*, 2019.

[45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[46] Luca Morreale, Noam Aigerman, Vladimir G. Kim, and Niloy J. Mitra. Semantic neural surface maps. In *Eurographics*, 2024.

[47] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017.

[48] OpenAI. Gpt-3.5 language model. `https://www.openai.com/research/gpt-3`, 2021. Accessed: May 21, 2023.

[49] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[50] 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, 2012.

[51] Gianluca Paravati, Fabrizio Lamberti, Valentina Gatteschi, Claudio Demartini, and Paolo Montuschi. Point cloud-based automatic assessment of 3d computer animation courseworks. *IEEE Transactions on Learning Technologies*, 10(4):532–543, 2016.

[52] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018.

[53] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial Functional Correspondence. In *Computer Graphics Forum*, 2016.

[54] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics*, 2022.

[55] Nicholas Sharp and Keenan Crane. A laplacian for nonmanifold triangle meshes. *Computer Graphics Forum*, 2020.

[56] Mingze Sun, Shiwei Mao, Puhua Jiang, Maks Ovsjanikov, and Ruqi Huang. Spatially and spectrally consistent deep functional maps. In *ICCV*, 2023.

[57] Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. Shape registration in the time of transformers. *Advances in Neural Information Processing Systems*, 34:5731–5744, 2021.

[58] Jinbao Wang, Shujie Tan, Xiantong Zhen, Shuo Xu, Feng Zheng, Zhenyu He, and Ling Shao. Deep 3d human pose estimation: A review. *Computer Vision and Image Understanding*, 2021.

[59] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[60] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3d: Few-shot 3d keypoint detection with back-projected 2d features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[61] Chengzhi Wu, Junwei Zheng, Julius Pfrommer, and Jürgen Beyerer. Attention-based point cloud edge sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5333–5343, 2023.

[62] Yuxin Yao, Bailin Deng, Weiwei Xu, and Juyong Zhang. Fast and robust non-rigid registration using accelerated majorization-minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[63] Tao Yu, Zerong Zheng, Kaiwen Guo, Jianhui Zhao, Qionghai Dai, Hao Li, Gerard Pons-Moll, and Yebin Liu. Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In *CVPR*, 2018.

[64] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *CVPR*, pages 6052–6061, 2021.

[65] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21769–21780, 2023.

[66] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017.

In this appendix, we provide more technical details and experimental results, including 1) A detailed description of the building blocks of our LG-Net in Sec. A; 2) Further qualitative results on matching heterogeneous shapes from SHREC'07 and DT4D-H in Sec. B.1; 3) Quantitative results following the setting from [34, 17, 25], where train/test with the sparse point clouds of fixed 1024 points in Sec. B.2; 4) Robustness evaluation of our method with respect to several perturbations in Sec. B.3; 5) More high-dimensional feature visualization and matching results of medical dataset in Sec. B.4; 6) Run-time analysis, hyper-parameter instruction in Sec. B.5 and Sec. B.6 respectively. Finally, the broader impacts are discussed in Sec. C.

A Technical Details

Fig. 5 depicts from left to right the architecture diagrams of our local attention, global attention, and fusion module.

Figs/network.png

Figure 5: The schematic illustration of the main blocks of our network.

B Experiments

B.1 Further Qualitative Results

Figs/DT4DSHREC07.png

Figure 6: We estimate correspondences between heterogeneous shapes from SHREC'07 and DT4D-H with DPC,SE-ORNET and one SSMSM, all trained on the SCAPE_r dataset. Our method outperforms the competing methods by a large margin.

In Fig. 6, we qualitatively visualize maps obtained by different methods tested in the SHREC'07 and DT4D-H benchmark. It is obvious that our results outperform all the competing the competing methods, showing superior generalization performance.

Table 7: Quantitative results on human and animals datasets. Acc signifies correspondence accuracy at 0.01 error tolerance, and err denotes average correspondence error. The **best** results in each column are highlighted.

Train	SHREC'19		SURREAL		TOSCA		SMAL	
Test	SHREC'19				TOSCA			
	acc ↑	err ↓	acc ↑	err ↓	acc ↑	err ↓	acc ↑	err ↓
3D-CODED[S] [23]	/	/	2.1%	8.1	/	/	0.5%	19.2
Elementary[S] [18]	/	/	0.5%	13.7	/	/	2.3%	7.6
CorrNet3D[U] [64]	0.4%	33.8	6.0%	6.9	0.3%	32.7	5.3%	9.8
DPC[U] [34]	15.3%	5.6	17.7%	6.1	34.7%	2.8	33.2%	5.8
SE-ORNet[U] [17]	17.5%	5.1	21.5%	4.6	38.3%	2.7	36.4%	3.9
HSTR[U] [25]	19.3%	4.9	19.4%	5.6	**52.3%**	1.2	33.9%	5.6
Ours [U]	**20.4%**	**4.7**	**23.4%**	**4.4**	43.7%	**1.0**	**37.9%**	**3.6**

Table 8: Mean geodesic errors ($\times 100$) on under different perturbations. Noisy PC means the input point clouds are perturbed by Gaussian noise. Rotated PC means the input point clouds are randomly rotated within ±30 degrees. The standard deviation value is shown in parentheses.

Method		Unperturbed	Noisy PC	Rotated PC
DiffFMaps[S] [42]	Trained on Mesh	12.0	14.9(2.57)	26.5(3.35)
NIE[W] [30]	Trained on Mesh	11.0	11.5(0.32)	19.9(1.29)
SSMSM[W] [11]	Trained on Mesh	4.1	5.4(0.11)	9.2(1.01)
DPC[U] [34]	Trained on PCD	17.3	18.2(0.80)	22.1(0.72)
SE-ORNet[U] [17]	Trained on PCD	24.6	24.7(0.15)	27.2(0.41)
Ours [U]		7.6	7.8(0.10)	8.7(0.60)

B.2 Further Quantitative Results

Sparse Humans/Animals Benchmarks: Following the prior works [34, 17, 25], we conduct the experiments with a consistent sampling point number of $n = 1024$. In addition to the datasets mentioned in the Sec. 4, two more datasets are included for training purposes including SURREAL and SMAL. **SURREAL** is the large-scale dataset from [23] comprises 230,000 training shapes, from which we select the first 2,000 shapes and use them solely for training. **SMAL** is from [66], which includes parameterized animal models for generating shapes. We employ the model to generate 2000 instances of diverse poses for each animal category, resulting in a training dataset comprising 10000 shapes.

Specifically, we train on the SURREAL and SHREC'19 dataset respectively, and then test on the SHREC'19 dataset. Similarly, we train respectively on SMAL and TOSCA dataset, and then test on the TOSCA dataset. As shown in Tab. 7, unlike HSTR[25], which achieves the best performance on its intra-dataset but lags behind SE-ORNet[17] on cross-dataset generalization, our approach excels in both intra-dataset and cross-dataset tests, surpassing all existing methods by over **4%**. This also complements Tab. 1 and Tab. 2, demonstrating that our method yields robust results whether trained/tested on dense or sparse point clouds.

B.3 Robustness

Moreover, we evaluate the robustness of our model with respect to noise and rotation perturbation and report in Tab. 8. More specifically, we perturb the point clouds by: 1) Adding per-point Gaussian noise with i.i.d $\mathcal{N}(0, 0.02)$ along the normal direction on each point; 2) Randomly rotating ± 30 degree along some randomly sampled direction. We perform 3 rounds of test, and report both mean error and the standard deviation in parentheses. Note that SE-ORNET[17] is designed for rotational robustness, which enjoys better rotation performance. Apart from that, our pipeline delivers the most robust performance among the baselines.

B.4 More Visualizations

High-dimensional feature visualization: To further validate the characteristics of the representations learned by our method, we present a set of more comprehensive visualizations of the features. As shown in Fig. 7, our feature distribution is more clean and localized. However, upon losing geometric or semantic information, the features across different dimensions become divergent, resulting in the loss of regular fine-grained representation at various levels.

Figs/feature.png

Figure 7: Visualization of different feature dimensions. **Dim.i** denotes the features of the $i-th$ dimension, where $i \leq 128$.

Matching results of medical dataset: To supplement Tab.5, we further visualize the matching results on the spleen organ in Fig.8, where excellent mapping is achieved regardless of whether the spleen exhibits various shapes or is positioned at different angles.

Figs/spleen.png

Figure 8: Matching result of the spleen dataset from [3].

More qualitative results: We further visualize the results of TOSCA, DT4D, and SCAPE-PV, which respectively serve as qualitative validation supplements for learning sparse point clouds in Tab. 7, the generalization capability in Tab. 2, and the adaptability to partial shapes in Tab. 3. The training and testing procedures align with the methods described in the aforementioned table, with quantitative supplements presented respectively in Fig. 4,Fig. 10 and Fig. 11, respectively.

B.5 Running Time

We perform all the experiments on a machine with NVIDIA A100-SMX4 80GB and Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz using the PyTorch 2.2.0 framework. Benchmarking on SCAPE_r dataset, our method achieves an average processing time of approximately **0.21** seconds per pair on testing, and approximately **2.35** seconds per iteration (6 batches) on training. In addition, it is also

Figs/horse.png

Figure 9: More qualitative results of TOSCA. All horse shapes from the dataset have been showcased.

Figs/dt4d.png

Figure 10: More qualitative results of DT4D. Our method demonstrates a notable improvement over other baselines.

Table 9: Hyper-parameters. The tables details the hyperparameter values that we used for the training of our method.

symbol	Description	Value
k	The nearest number for computing geometrically similarity loss	500
k_{attn}	The number for searching latent nearest features in local attention	40
k_c	The number for self-construction/cross-construction neighborhood size	40
k_m	Mapping loss neighborhood size	40
k'_{attn}	The number for local attention when training point cloud with fixed $n = 1024$ points	10
k'_m	The mapping loss neighborhood size when training point cloud with fixed $n = 1024$ points	10
k'_c	The number for construction neighborhood size when training point cloud with fixed $n = 1024$ points	40
α	Mapping loss neighbor sensitivity	8
TEs	Training epochs	50
H, W	The size of our projected image	224,224

feasible to train on a single NVIDIA GeForce RTX 3090 24GB, only necessitating a reduced batch size.

B.6 Additional Hyper-parameter Details

For a comprehensive understanding of the specific hyper-parameter configurations, please refer to Tab. 9.

Figs/scape_pv.png

Figure 11: Qualitative results of SCPAE-PV.

C Broader Impacts

We fail to see any immediate ethical issue with the proposed method. On the other hand, since our method is extensively evaluated in matching human shapes and achieves excellent results, one potential misuse can be surveillance, which may pose negative societal impact.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have made the main claims in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in the end of paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper mainly focuses on methods and their applications which does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the Implementation details in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will release the code after acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided the Implementation details in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have reported the robustness of our method regarding rotation perturbation in Tab. 8.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported computational resources in Sec. B.5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Of course.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed broader impacts in Sec. C

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not see such danger.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have carefully checked and confirmed this.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: There is no new assets provided.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work is irrelavent to crowd-sourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work is irrelavant to this front.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.