SE-ORNet: Self-Ensembling Orientation-aware Network for Unsupervised Point Cloud Shape Correspondence

Jiacheng Deng[1,*], Chuxin Wang[1,*], Jiahao Lu[1], Jianfeng He[1], Tianzhu Zhang[1,3,†], Jiyang Yu[2], Zhe Zhang[3]
[1]University of Science and Technology of China, [2]China Academy of Space Technology, [3]Deep Space Exploration Lab
{dengjc, wcx0602, lujiahao, hejf}@mail.ustc.edu.cn, tzzhang@ustc.edu.cn
yujiyang@spacechina.com, cnclepzz@126.com

Abstract

Unsupervised point cloud shape correspondence aims to obtain dense point-to-point correspondences between point clouds without manually annotated pairs. However, humans and some animals have bilateral symmetry and various orientations, which lead to severe mispredictions of symmetrical parts. Besides, point cloud noise disrupts consistent representations for point cloud and thus degrades the shape correspondence accuracy. To address the above issues, we propose a Self-Ensembling ORientation-aware Network termed SE-ORNet. The key of our approach is to exploit an orientation estimation module with a domain adaptive discriminator to align the orientations of point cloud pairs, which significantly alleviates the mispredictions of symmetrical parts. Additionally, we design a self-ensembling framework for unsupervised point cloud shape correspondence. In this framework, the disturbances of point cloud noise are overcome by perturbing the inputs of the student and teacher networks with different data augmentations and constraining the consistency of predictions. Extensive experiments on both human and animal datasets show that our SE-ORNet can surpass state-of-the-art unsupervised point cloud shape correspondence methods.

1. Introduction

With the cost of LiDAR and depth cameras falling, it is more accessible to obtain 3D point cloud data. For real-world applications, such as articulated motion transfer [5, 26] and non-rigid human body alignment [3], the correspondence between two point clouds is indispensable. However, we are hard to directly obtain the correspondence between two raw point clouds due to various object orientations and ununified coordinate systems.

To accurately find the point-to-point correspondence between two point clouds, spectral-based methods [1, 9, 15, 19, 28] have been proven as practical shape correspondence methods by computing functional mapping between the projected features and learning a transformation for the correspondence. Nevertheless, the spectral-based methods suffer from complicated pre-processing steps and the necessity for connectivity information between points. With the rapid development of deep learning, many fully supervised point cloud shape correspondence methods [4, 8, 16] have been proposed to lead to remarkable progress. However, these methods rely on a large amount of carefully annotated point cloud pairs, which are expensive and time-consuming to collect. To relieve the annotation cost of fully supervised methods, unsupervised methods [12, 32] that utilize unlabeled data for model training have attracted more and more attention. CorrNet3D [32] proposes the first unsupervised deep learning framework for building dense correspondence between point clouds in an end-to-end manner. DPC [12] models the local point cloud structure by exploring the proximity of points using DGCNN [29] and designs reconstruction losses to extract continuous point cloud rep-

Figure 1. **The visualization of dense point matching results.** Three point cloud pairs have different relative rotation angles. GT denotes ground truth. The correspondence is visualized by transferring colors from source to target according to matching results. The baseline predicts many false matches, especially for symmetrical, similar parts of the object. Our method achieves accurate matches for these parts with our orientation estimation module.

*Equal Contribution
†Corresponding Author

resentations. However, in the scanning process of 3D scanner, due to light, vibration and other factors, point cloud noise will be inevitably generated. Meanwhile, the preprocessing of point cloud (such as random subsampling) will also introduce noise. Unfortunately, the previous methods fail to adequately consider the point cloud noise, which negatively impacts the point cloud representations. Besides the noise, existing methods lack attention to symmetrical parts of the body. The mismatching issue of symmetrical parts is challenging in this task, which was also spotted by the previous method [32] but has yet to be solved.

By studying the previous point-based shape correspondence methods [4, 8, 12, 16, 32], we summarize two key issues that need consideration to achieve a more accurate shape correspondence: 1) *How to overcome the disturbance of point cloud noise to get robust and consistent point cloud representations?* Point cloud noise perturbs the spatial coordinates of point cloud and interferes with local structure modeling. Therefore, it is necessary to overcome noise disturbances. 2) *How to solve the mismatching issue of symmetrical parts in point clouds with different body orientations?* As shown in Figure 1, for the pair of bilaterally symmetrical human point clouds facing the opposite directions, existing methods predict the completely reverse and seriously wrong point cloud correspondence due to the similar structure and position. The specific relative rotation angles lead to severe mispredictions of symmetrical parts.

To achieve the above goals, we propose a *Self-Ensembling Orientation-aware Network* (SE-ORNet) for unsupervised point cloud shape correspondence. We integrate orientation modeling and consistent point cloud representations under a unified self-ensembling framework, which consists of a pair of teacher and student models, an orientation estimation module, and an adaptive domain discriminator. Firstly, we design a new augmentation scheme to produce augmented samples with rotation and Gaussian noise, and record the rotation angles as rotation angle labels. Then, we formulate soft labels and consistency losses to encourage consensus among ensemble predictions of augmented and raw samples, aiming to perceive the difference in body orientation and overcome the point cloud noise disturbance to obtain consistent point cloud representations. In addition, we design a plug-and-play lightweight Orientation Estimation Module, which aligns the orientations of two point clouds to solve the mismatching issue of symmetrical parts in point clouds. Without the real label of relative rotation angle between the source and target, we supervise the training with the rotation angle labels and calculate angle losses. However, there is a noticeable domain gap between the rotation-augmented samples and the real samples. Therefore, we design a discriminator to achieve domain adaptation and calculate the domain losses. Furthermore, the discriminator facilitates the Orientation Estimation Module to mine the valuable knowledge in the rotation-augmented samples to compensate for the information loss of the real relative rotation angles.

In summary, the main contributions of this work are as follows: (i) We design a plug-and-play lightweight Orientation Estimation Module that accurately aligns the orientations of point cloud pairs to achieve correct matching results of symmetrical parts. (ii) We integrate point cloud orientation modeling and consistent point cloud representation learning with the disturbance of point cloud noise into a unified self-ensembling framework. (iii) Our method attains state-of-the-art performance on both human and animal benchmarks, and extensive experimental results verify the superiority of our designs.

2. Related Work

In this Section, we give a brief overview of related works on point cloud shape correspondence, including learning on point clouds, shape correspondence, and self-ensembling approaches.

Learning on Point Clouds. PointNet [21] learns from global information through multi-layer perceptrons and max-pooling operation. PointNet++ [22] devises a hierarchical architecture that recursively partitions the point cloud to extract local features more effectively. Recent works explore local aggregators via relations [23, 31, 33], and graphs [29, 34]. PointCNN [13] transforms neighboring points to the canonical order to apply traditional convolution on point clouds. DGCNN [29] creates a graph in the feature space and designs EdgeConv [29] to learn the edge features of the graph in each layer. However, the methods are commonly based on some assumptions of implicit local geometry, which may result in sensitivity to point cloud disturbances.

Shape Correspondence. As an active research area in computer vision and graphics, point cloud shape correspondence methods roughly include spectral-based methods [1,9,15,19,28] and point-based methods [4,8,12,16,32]. Spectral-based methods require connectivity information to compute the LBO eigenvectors as basis functions and infer a linear transformation for shape correspondence. However, with regard to point cloud data, connectivity information is difficult to obtain directly while point-based methods directly process point clouds without connectivity information to find the dense point-to-point mapping between two point clouds with deformable 3D shapes. Deprelle et al. [4] propose representing shapes as the deformation and combination of learnable elementary 3D structures. Groueix et al. [8] employ an encoder-decoder architecture to obtain and constrain the similarity matrix with manually annotated labels. The deep learning methods train their neural networks in a data-driven manner and improve performance to a large extent. However, manually labeling the point-to-point cor-

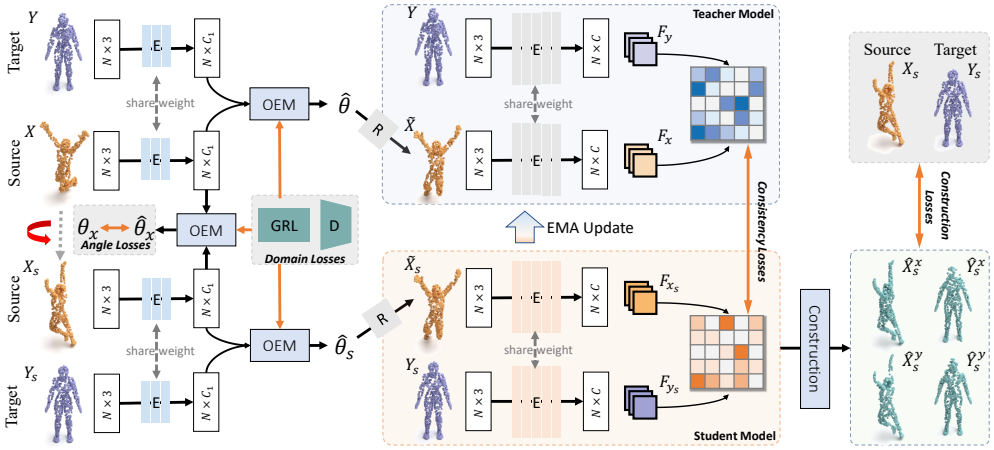

Figure 2. **The overview of our self-ensembling orientation-aware network for unsupervised point cloud shape correspondence.** X_s is generated from the raw source point cloud X by random rotation and Gaussian noise, while Y_s is only augmented by Gaussian noise. We design the Orientation Estimation Module to estimate the rotation θ of the source with respect to the target and align the point cloud in position space. Afterward, the aligned point cloud pairs are input to the teacher and student models, respectively, and the correspondence is predicted through a DGCNN backbone. Finally, we supervise the student model by the consistency losses and the construction losses, and the teacher model updates the parameters using the exponential moving average (EMA) strategy. The gradient reversal layer (GRL) acts as the identity function during forward propagation, but is multiplied by -1 during backward propagation.

respondence between two point clouds in 3D space takes much time and effort. Therefore, some unsupervised point cloud shape correspondence methods [12, 32] are proposed to reduce the overhead of labeling. CorrNet3D [32] is the first unsupervised deep learning framework. DPC [12] designs several reconstruction losses to smooth point cloud representations. Due to the lack of annotation, the mismatching issue of symmetrical parts in point clouds with different orientations has become an undeniable problem in unsupervised shape correspondence area.

Self-ensembling Approaches. Self-ensembling approaches improve the model generalization by encouraging consensus among ensemble predictions of unknown samples with small perturbations. Γ model [24] consists of two identical parallel branches that respectively take raw images and corrupted images as inputs. In contrast to Γ model, Π model [11] integrates two parallel branches into a single branch. As an extension of the Π model, the temporal model [11] forces the consistency between the outputs and the aggregated outputs over previous training epochs. Mean Teacher [27] replaces network prediction average with network parameter average via the exponential moving average (EMA) strategy. We design a framework similar to Mean Teacher and adapt it for the unsupervised point cloud shape correspondence task. The proposed framework facilitates the network to yield consistent and accurate predictions under noise perturbations and orientation rotations.

3. Method

3.1. Overview

The unsupervised point cloud shape correspondence is to find the mapping $f : X \rightarrow Y$ between two point clouds (source X and target Y) without ground-truth correspondence annotations. Figure 2 illustrates the pipeline of our approach. Given source X and target Y point clouds, we utilize random rotation and Gaussian noise to generate the augmented source point cloud X_s, while we augment Y by Gaussian noise to obtain Y_s. Due to the absence of the relative rotation angle labels θ, θ_s, we use the rotation angle labels θ_x to guide the Orientation Estimation Module learning and transfer the valuable information by the adversarial domain adaptation method. The aligned point cloud pairs are input to the teacher and student models, respectively, and we use the DGCNN [29] backbone to output the similarity matrices. After that, we generate two reconstructed point clouds $\hat{X}, \hat{Y}$ based on the predicted similarity matrices by the student model. Finally, the student model is supervised by the consistency losses and the construction losses, while the parameters of teacher model are updated by the exponential moving average (EMA) strategy.

3.2. Orientation Estimation

We provide the overview of the orientation estimation in Figure 3. The encoded features P^s_{in}, P^t_{in} are fed into the Feature Interaction Module (FIM) for feature fusion. The fused features P_{out} are enhanced by a single-layer edgeconv. To predict the relative orientation, the features $\hat{P}_{in}$ are fed into the classification head. We also input $\hat{P}_{in}$ into the discriminator to determine whether the pair comes from the same shape or from different shapes.

Feature Interaction Module. As shown in Figure 4, the feature interaction module is a query-based graph convolution. The point features in the source point cloud are updated by querying points with similar features in the tar-

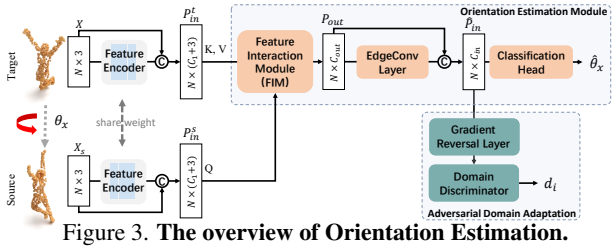

Figure 3. **The overview of Orientation Estimation.**

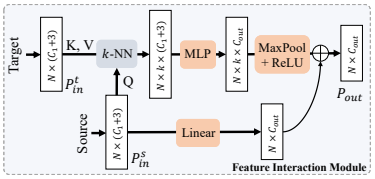

Figure 4. **The Feature Interaction Module (FIM).**

get point cloud. Let P_{in}^s, P_{in}^t be the inputs of the feature interaction module. For each point $p_i \in P_{in}^s$, we first consider it as a query to search for the k-nearest neighbors $q_i^j, 1 \leq j \leq k$ in the target point cloud P_{in}^t with respect to the Euclidean distance defined in the C_1-dimensional feature space. To better model the relative rotation relationships in space, we use spatial position differences and feature differences as features of each edge, denoted as $(p_i, q_i^j - p_i)$, where $p_i, q_i^j \in \mathcal{R}^{(C_1+3)}$. Then we use a multilayer perceptron (MLP) to compute a new feature $e_{ji} = \text{MLP}(p_i, q_i^j - p_i)$ from each edge. For each point, we aggregate its k edge features into a new point feature through Max pooling and ReLU activation. In addition, we add a linear layer that skip-connects the output P_{out} with the input to make the block residual.

Rotation Classification Head. We concatenate the output feature P_{out} and the enhanced feature as the input $\hat{P}_{in}$ of the Rotation Classification Head. Inspired by the orientation prediction in the point cloud detection methods [18,20], we consider the prediction of relative rotation angle as a classification task. That coarsely aligning the orientations of the source and target point clouds is enough to solve the problem of mismatching issue of symmetrical parts in point clouds. Thus, we pre-define M equally divided angle bins and then use an MLP head to classify the relative rotation angle into those pre-defined categories. Specifically, we compress $\hat{P}_{in}$ using maximum pooling and average pooling to obtain the global features and then predict the probability distribution of the relative rotation angle. Finally, we choose the angle bin with the highest probability as the relative rotation angle of the source and target point clouds.

Adversarial Domain Adaptation. Due to the absence of relative rotation angle θ, θ_s, we utilize the relative rotation angle θ_x to guide the Orientation Estimation Module learning. However, there is a noticeable domain gap between the rotation-augmented samples and the real samples. To eliminate the domain gap, we use a discriminator to identify whether the input features $\hat{P}_{in}$ of the classification head are from the rotation-augmented samples or the real samples. Specifically, we use a PointNet-like module to process the features $\hat{P}_{in}$ and predict the probability d_i that the P_{in} is from the real samples:

$$d_i = \text{MLP}_2\left\{\max_N\left\{\text{MLP}_1\left[\hat{P}_{in}\right]\right\}\right\}, \tag{1}$$

where max refers to the global Max pooling. By using the discriminator, we train the model to disregard the distinction between the two domains and instead prioritize learning the relative rotation information.

Relative Rotation Angle Supervision. The angle loss is computed with a cross entropy loss:

$$\mathcal{L}_{angle} = -\sum_{i=1}^{M} \theta_x^i \log\left(\hat{\theta}_x^i\right), \tag{2}$$

where M is the number of the pre-defined angle bins, $\hat{\theta}_x^i$ and θ_x^i denote the predicted probability distribution of relative rotation angle and the rotation angle label, respectively. The domain loss is computed with a Focal Loss [14]:

$$\mathcal{L}_{domain} = -\left(1 - d_i\right)^{\gamma} \log\left(d_i\right), \gamma > 1, \tag{3}$$

where $\mathcal{L}_{domain}$ denotes the domain loss and γ denotes the tunable focusing parameter.

3.3. Self-Ensembling Framework

As shown in Table 7, the point cloud orientation and Gaussian noise are two crucial factors in the unsupervised point cloud shape correspondence task. The existing methods ignore noise interference on the point cloud correspondence, making it difficult to obtain robust point cloud representations. To address the above problems, we utilize the Mean Teacher architecture [27] and design two consistency losses to constrain the student model for robust feature representations under orientation disturbance and Gaussian noise interference.

Stochastic Transform. We apply stochastic transformations that include rotation and Gaussian noise on the point clouds for the student network formulated as $\tau = (\mathcal{R}, \mathcal{N})$. More specifically, given the raw point cloud pair (X, Y), we first apply Gaussian noise to the target point cloud Y:

$$y_i^s = y_i + n_i, n_i \sim \mathcal{N}(0, \sigma^2), \tag{4}$$

where $\mathcal{N}(0, \sigma^2)$ means the Gaussian distribution with the 0 mean and σ standard deviation. Then, we utilize Gaussian noise and random rotation along the vertical z-axis to augment the source point cloud X:

$$x_i^s = \mathcal{R}(\theta_x) \odot x_i + n_i, n_i \sim \mathcal{N}(0, \sigma^2), \tag{5}$$

where $\mathcal{R}$ denotes the rotation around the z-axis and θ_x is the rotation angle sampled from $[0, 2\pi]$.

Architecture of Teacher & Student Models. Given the raw and augmented point cloud pairs $(X, Y), (X_s, Y_s)$, we use the Orientation Estimation Module to align source and target orientations. Then, our approach follows the Mean Teacher paradigm [27] and inputs the aligned point cloud pairs $(\tilde{X}_s, Y_s), (\tilde{X}, Y)$ into the student and teacher models, respectively. We use a variant of DGCNN [29] backbone to embed the aligned source $\tilde{X}$ and target Y point clouds to a high-dimensional feature space as $F_x, F_y \in \mathcal{R}^{N\times C}$. Unlike DGCNN, which uses dynamic graphs to select neighbors, we follow [12] to use static graphs for the neighbors. Specifically, we select the k-nearest neighbors in the coordinate space instead of the feature space. Then, we use the cosine similarity S_{xy} of source and target point features F_x, F_y to measure their correspondence.

$$s_{ij} = \frac{F_x^i \cdot (F_y^j)^T}{\|F_x^i\|_2 \|F_y^j\|_2}, \tag{6}$$

where F_x^i, F_y^j are the i$^{\text{th}}$ and j$^{\text{th}}$ rows of F_x, F_y, respectively, and $(\cdot)^T$ denotes a transpose operation.

Soft Label & Consistency Loss. To enhance the perception of local structures and the robustness of the model, we design two consistency losses to maintain feature consistency under Gaussian noise and orientation disturbances. We utilize the cross-similarity S_{xy}^t obtained from the teacher model as a soft label and the $\text{Smooth}\, L_1$ loss to constrain the cross-similarities $S_{x_s y_s}^s$ of the student model:

$$\mathcal{L}_{ccs} = \text{Smooth}\, L_1(S_{xy}^t, S_{x_s y_s}^s), \tag{7}$$

where $\mathcal{L}_{ccs}$ denotes the consistency loss of the cross-similarities. To reduce the interference of point cloud orientation on correspondence estimation, we model the feature consistency of the strongly augmented source X_s and raw source X. We constrain the consistency of the self-similarities S_{xx} to ensure the consistency of neighboring points with similar features. We use the feature embedding F_x of the source point cloud to compute the self-similarity S_{xx} by Equation (6). Then, we use the $\text{Smooth}\, L_1$ loss to constrain the self-similarities $S_{x_s x_s}^s$ of the student model with the self-similarities S_{xx}^t from the teacher model:

$$\mathcal{L}_{css} = \text{Smooth}\, L_1(S_{xx}^t, S_{x_s x_s}^s), \tag{8}$$

where $\mathcal{L}_{css}$ denotes the consistency loss of the self-similarities. The above two consistency losses ensure the robustness of feature embedding under Gaussian noise and orientation disturbances for accurate correspondence.

3.4. Model Training & Inference

In addition to the above mentioned consistency losses, angle loss, and domain loss, we use reconstruction losses to promote a unique point matching between the shape pair. Following the previous work [12], we perform the cross-construction operation to construct the target shape $\hat{Y}$ by using the feature similarity S_{xy} between source and target point clouds, and the target point coordinates Y as follows:

$$\hat{y}_{x_i} = \sum_{j\in\mathcal{N}_\mathcal{Y}(x_i)} \frac{e^{s_{ij}}}{\sum_{l\in\mathcal{N}_\mathcal{Y}(x_i)} e^{s_{il}}} y_j, \tag{9}$$

where $\mathcal{N}_\mathcal{Y}(x_i)$ is latent k-nearest neighbors of x_i in the target Y. When the source and target point clouds are identical, we refer to the construction operation as self-construction. As shown in Figure 2, we obtain the point clouds $\hat{Y}_s^x, \hat{X}_s^x, \hat{X}_s^y, \hat{Y}_s^y$ by cross-construction and self-construction. Then, we constrain the training with the construction loss as follows:

$$\begin{aligned}\mathcal{L}_{cons} =& \lambda_{cc}(\text{CD}(Y_s, \hat{Y}_s^x) + \text{CD}(X_s, \hat{X}_s^y)) \\ &+ \lambda_{sc}(\text{CD}(Y_s, \hat{Y}_s^y) + \text{CD}(X_s, \hat{X}_s^x)),\end{aligned} \tag{10}$$

where $\lambda_{cc}, \lambda_{sc}$ are hyperparameters and CD means the Chamfer Distance. Finally, we add a regularization term to correspond close points in the source to close points in the target.

$$\mathcal{L}_{norm} = \sum_i \sum_{l\in\mathcal{N}_\mathcal{Y}(x_i)} e^{\|x_i - x_l\|_2^2/\alpha} \|\hat{y}_{x_i} - \hat{y}_{x_l}\|_2^2, \tag{11}$$

where $\mathcal{N}_\mathcal{Y}(x_i)$ is the same as defined in Equation 9 and α is a hyperparameter. To sum up, the total loss of our unsupervised point cloud shape correspondence method is:

$$\begin{aligned}\mathcal{L}_{total} =& \lambda_1\mathcal{L}_{ccs} + \lambda_2\mathcal{L}_{css} + \lambda_3\mathcal{L}_{angle} \\ &+ \lambda_4\mathcal{L}_{domain} + \mathcal{L}_{cons} + \lambda_5\mathcal{L}_{norm},\end{aligned} \tag{12}$$

where λ_i is a hyperparameter, balancing the contribution of different loss terms. During inference, we set the closest point y_j^* in the feature space for each point x_i as its corresponding point. This selection rule can be formulated as:

$$f(x_i) = y_{j^*}, j^* = \underset{j}{\text{argmax}}(s_{ij}). \tag{13}$$

4. Experiments

4.1. Experimental Setup

Dataset. To demonstrate the effectiveness and generalization of our method, we perform experiments on human and animal datasets. We conduct experiments on human datasets according to DPC's [12] scheme. For the large-scale dataset, we randomly downsample the SURREAL [8] dataset, which contains 230000 training shapes, into 2000 shape pairs as the training set. For the test set, we use the SHREC'19 [17] dataset, which contains 44 real human models, and pair them into 430 annotated test examples. To further verify the ability of our method to learn discriminative feature expression with a small data size, we train

SE-ORNet on the pairs randomly sampled from 44 SHREC instances, and the testing is still conducted on the official 430 SHREC'19 pairs.

For animal datasets, we also conduct experiments with different dataset scales. We use the large-scale SMAL [35] dataset and TOSCA [2] dataset as the training set and test set, respectively. SMAL dataset consists of parameterized models of various animals, and we randomly sample SMAL to obtain the corresponding shape pairs as the training set. TOSCA is generated by deforming three template meshes (human, dog, and horse) into multiple poses. We pair the 41 animal figures in TOSCA from the same category to form a training set of 260 samples and a test set of 286 samples. Because the number of points in different shapes varies, we make a random downsample of the original point cloud to a fixed number $n = 1024$, as done in CorrNet3D [32].

Evaluation Metrics. The evaluation metrics include the average correspondence error and the correspondence accuracy. Based on the Euclidean-based measure, the average correspondence error is defined for a pair of source and target shapes (X, Y) as follows:

$$err = \frac{1}{n} \sum_{x_i \in X} \left\| f\left(x_i\right) - y_{gt} \right\|_2 , \tag{14}$$

where $y_{gt} \in Y$ is the ground-truth matching point to x_i. The unit is centimeter(cm). And the correspondence accuracy can be formulated as:

$$acc(\epsilon) = \frac{1}{n} \sum_{x_i \in X} \mathbb{I}\left(\left\| f\left(x_i\right) - y_{gt} \right\|_2 < \epsilon d \right) , \tag{15}$$

where $\mathbb{I}(\cdot)$ is the indicator function, d is the maximal Euclidean distance between points in Y, and $\epsilon \in [0, 1]$ is an error tolerance.

Implementation Details. For a fair comparison with existing methods [12, 32], we use the same DGCNN [29] backbone with four EdgeConv blocks as the feature extractor in the self-ensembling framework. The standard deviation σ in Equation (4) is set as 0.1 for human datasets and 0.15 for animal datasets. In the Orientation Estimation Module, the feature encoding module is a simplified DGCNN with three EdgeConv [29] blocks whose layer output sizes are 64, 128, and 256. The k of k-NN is set as 24, and the slope of all LeakyReLU is 0.2. We feed the output of the last layer into the proposed feature interaction module with the output size 256. Then we refine the feature by an EdgeConv layer with the same output size. The angle classification head consists of three Linear-BN-ReLU and an output Linear. The channels are 256, 128, and 128 for the three Linear-BN-ReLU. The last Linear outputs the probability for classification. We set the number of bins M as 8 in the angle classification head, where each bin represents a range of $45°$. The domain discriminator is a PointNet-like module consisting of two MLPs in Equation (1). The channels of MLP_1 are 512, 256, and 128, while the channels of MLP_2 are 256, 128, and 256. We follow [12] and use a neighborhood size $k = 10$ in Equation (9) and (11). $\lambda_c c$ and $\lambda_s c$ in Equation (10) are set as 1 and 10, respectively. λ_1, λ_2, λ_3, λ_4, and λ_5 in Equation (12) are set as 0.1, 0.1, 1.0, 0.8, and 1.0, respectively.

Table 1. **Comparison on SHREC and SURREAL benchmarks.** Here, acc means the correspondence accuracy at an error tolerance of 0.01, while err refers to the average correspondence error. Higher accuracy and lower error reflect a better result.

Method	Input	SHREC		SURREAL	
		acc ↑	err ↓	acc ↑	err ↓
Diff-FMaps [16]	Point	/	/	4.0%	7.1
3D-CODED [8]	Point	/	/	2.1%	8.1
Elementary [4]	Point	/	/	2.3%	7.6
CorrNet3D [32]	Point	0.4%	33.8	6.0%	6.9
DPC [12]	Point	15.3%	5.6	17.7%	6.1
Ours	Point	**17.5%**	**5.1**	**21.5%**	**4.6**

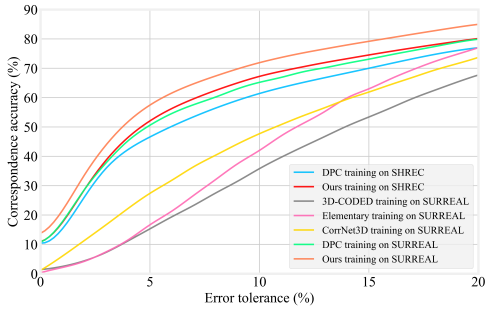

Figure 5. **Correspondence accuracy at various error tolerances for human datasets.** The methods are trained on the SHREC or SURREAL dataset and evaluated on SHREC'19 test pairs. Compared with other methods, our approach achieves an impressive performance improvement.

4.2. Comparison on Human Datasets

For a fair comparison with existing methods, we do not use any post-processing or additional connectivity information. In addition, we follow DPC [12] and train our model on the SURREAL and SHREC datasets, respectively. Then we test our model on the official 430 SHREC'19 pairs.

Evaluation on SHREC dataset. As shown in Table 1, our approach shows significant performance improvements on the SHREC benchmark and achieves new SOTA performance by 2.2% improvements in accuracy and 0.5 reductions in error. To show the improvement under different error tolerances, we present the correspondence accuracy for point-based methods trained on SHREC and evaluated on the SHREC'19 test set. As shown in Figure 5, our method achieves better results with different error tolerances.

Cross-dataset Generalization. In Table 1 and Figure 5, we also report the comparison with other methods on the SURREAL benchmark. The models are trained on the SURREAL dataset and evaluated on the SHREC'19 test

Table 2. **Comparison on TOSCA and SMAL benchmarks.** Here, acc means the correspondence accuracy at an error tolerance of 0.01, while err refers to the average correspondence error. Higher accuracy and lower error reflect a better result.

Method	TOSCA		SMAL	
	acc ↑	err ↓	acc ↑	err ↓
3D-CODED [8]	/	/	0.5%	19.2
Elementary [4]	/	/	0.5%	13.7
CorrNet3D [32]	0.3%	32.7	5.3%	9.8
DPC [12]	34.7%	2.8	33.2%	5.8
Ours	**38.2%**	**2.7**	**36.4%**	**3.9**

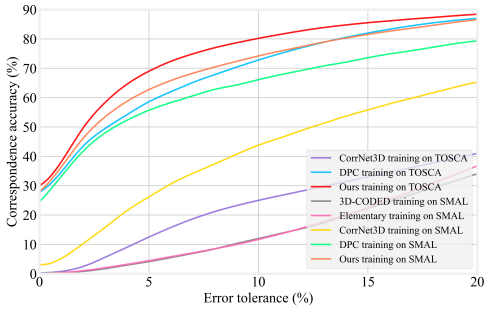

Figure 6. **Correspondence accuracy at various error tolerances for animal datasets.** The methods are trained on the TOSCA or SMAL dataset and evaluated on the official TOSCA test pairs. Compared with other methods, our method achieves a desirable performance improvement.

set. The large-scale training set of the SURREAL dataset helps the deep learning-based methods perform better, even though there is a domain gap between the SURREAL and SHREC datasets. With our proposed method, the correspondence accuracy reaches 21.5% at an error tolerance of 0.01, and the average correspondence error is reduced to 4.6 on the SHREC'19 test set.

4.3. Comparison on Animal Datasets

To verify the adaptability of our method to point clouds of different shapes, we conduct experiments on two animal benchmarks. Similar to human datasets, we train our method on TOSCA and SMAL datasets respectively, and test on the TOSCA test dataset. Table 2 and Figure 6 show the competitive results on TOSCA and SMAL benchmarks. Our method achieves a 3.5% accuracy improvement on the TOSCA benchmark and a 3.2% accuracy improvement on the SMAL benchmark. Compared with the human datasets, the animal datasets have various shapes with aligned orientations. Thus, the above performance gains come mainly from our self-ensembling framework, which can learn reliable features on complex data. To verify the effect of the Orientation Estimation Module, we test our method and baseline using different augmented test sets in Section 4.6.

4.4. Comparison on Real-world Dataset

CMU Panoptic [10] is a dataset of scanned point clouds of human subjects in various poses, containing noise, outliers, occlusions, and clutter. Meanwhile, SHREC'20 [6] dataset contains real scans of various four-legged animal models. As shown in Table 3 and Table 4, we provide the results under two real scan datasets, demonstrating the remarkable robustness of our model to noise.

Table 3. **Comparison on CMU-Panoptic benchmark.** Here, err means average Euclidean keypoint error (cm).

	3D-CODED [8]	DIF-Net [25]	CorrNet [32]	IFMatch [7]	**Ours**
err	17.1	15.3	14.8	8.5	**3.2**

Table 4. **Comparison on SHREC'20 benchmark.** The training dataset is indicated in the bracket.

Method	SHREC'20 [SURREAL]		SHREC'20 [SMAL]	
	acc ↑	err ↓	acc ↑	err ↓
DPC [12]	25.0%	3.2	24.5%	7.5
Ours	**29.9%**	**1.2**	**25.4%**	**2.9**

Table 5. **Evaluation of the model with different designs on SURREAL.** $\mathcal{L}_{css}$ is the consistency loss of the self-similarities, $\mathcal{L}_{ccs}$ is the consistency loss of the cross-similarities, τ means the stochastic transform, OEM means we use the Orientation Estimation Module, FIM is the Feature Interaction Module, and DAM is the Domain Adaptation Module.

τ	$\mathcal{L}_{ccs}$	$\mathcal{L}_{css}$	OEM	FIM	DAM	SURREAL	
						acc ↑	err ↓
✗	✗	✗	✗	✗	✗	17.7%	6.1
✓	✓	✗	✗	✗	✗	18.8%	5.7
✓	✓	✓	✗	✗	✗	19.2%	5.6
✓	✓	✓	✓	✗	✗	19.5%	5.5
✓	✓	✓	✓	✓	✗	20.4%	5.1
✓	✓	✓	✓	✓	✓	**21.5%**	**4.6**

4.5. Ablation Study

Evaluation of the model with different designs. In this section, we perform extensive ablation studies on the SURREAL dataset to evaluate the effectiveness of each design. Table 5 demonstrates the performance of the model with different designs. Specifically, the first line is the results of DPC [12], which is our baseline model. The second row indicates that the self-ensembling framework with a stochastic transform achieves a better performance than the original model. By using $\mathcal{L}_{css}$ to constrain the consistency of source features before and after augmentation, the correspondence accuracy can be improved by 0.4%, as shown in the third row. As shown in the fourth row, adding the Orientation Evaluation Module without the Feature Interaction Module and the Domain Adaptation Module, the performance has a slight improvement. In the fifth row, by introducing the Feature Interaction Module into the Orientation Evaluation Module, the correspondence accuracy can be improved by 0.9%. Finally, after utilizing the Domain Adaptation Module, the correspondence accuracy is improved by 1.1%.

Effect of the Self-Ensembling Framework. As shown in Table 6, we modify the self-ensembling framework to show the effect of each designed component. Removing either of the consistency losses leads to a drop in performance, which indicates that constraining the student net-

Table 6. **Effect of the Self-Ensembling Framework on SURREAL.** Here, we modify the self-ensembling framework to show the effect of each designed component. $\mathcal{N}$ means Gaussian noise, and $\mathcal{R}$ means random rotation along the vertical z-axis.

SURREAL	w/o $\mathcal{L}_{ccs}$	w/o $\mathcal{L}_{css}$	w/o $\mathcal{N}$	w/o $\mathcal{R}$
acc ↑	19.9%	20.3%	20.6%	18.8%
err ↓	5.3	5.1	4.9	5.7

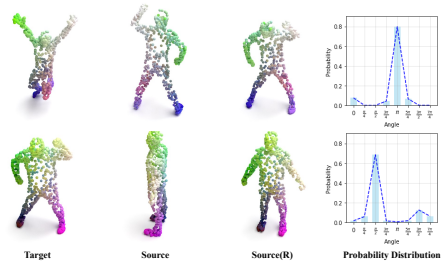

Figure 7. **Effect of the Orientation Estimation Module.** We visualize of the point clouds before and after orientation rectification on SHREC'19 test set. We denote the raw source point cloud as Source and the rectified one as Source(R). Besides, we provide the probability distributions of the relative rotation angle prediction.

Table 7. **Robustness Analysis.** To verify the robustness of our method, we test our method and baselines using different augmented test sets. $\mathcal{N}$ means Gaussian noise with standard deviation σ of 0.1 and $\mathcal{R}$ means random rotation along the vertical z-axis.

$\mathcal{N}$	$\mathcal{R}$	SURREAL(B)		SURREAL		SMAL(B)		SMAL	
		acc ↑	err ↓	acc ↑	err ↓	acc ↑	err ↓	acc ↑	err ↓
✗	✗	17.47%	6.30	21.55%	4.65	33.79%	5.78	36.39%	3.88
✓	✗	14.38%	8.74	21.55%	4.66	30.21%	6.06	36.15%	3.92
✗	✓	8.99%	9.33	17.59%	5.81	9.98%	12.24	28.16%	5.63
✓	✓	7.80%	11.28	17.58%	5.79	9.54%	12.80	27.60%	5.89

work with soft labels facilitates the consistency of the point cloud representations. When Gaussian noise and rotation augmentation are removed, the model performances show different degrees of degradation. The above experiments illustrate our self-ensemble method can obtain a more robust feature representation of the point cloud through data augmentation and consistency losses.

Effect of the Orientation Estimation Module. To verify the effect of the proposed Orientation Estimation Module, we visualize the point clouds after orientation rectification and the probability distributions of the relative rotation angle predictions. As shown in Figure 7, our method can accurately estimate the relative rotation angle and align the point cloud orientation of the source with that of the target. More results refer to the supplementary materials.

4.6. Robustness Analysis

To verify the robustness of our method, we use different augmentations on the test set. As shown in Table 7, we use Gaussian noise and random rotation for data augmentation of the test sets. Compared to our method, the baseline shows a significant performance decrease for test sets

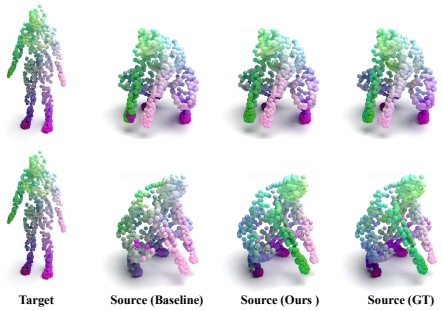

Figure 8. **Visualization of the correspondence results with rotation augmentation.** With rotation augmentation on point cloud pairs, the baseline shows regrettable performances, while our method still retains desirable performances.

Figure 9. **Visualization of point cloud pairs with deformations on real scanned Owlii dataset.** In each pair, the left one is the source and the right one is the target.

with Gaussian noise. For test sets with random rotation, the baseline performance is greatly degraded, while our method retains an acceptable performance. Figure 8 shows that our SE-ORNet can handle the orientation inconsistency issue of source and target well. To further verify the robustness and generalization of our method, we conduct experiments on the real scanned Owlii dataset [30] and present the visualization in Figure 9. The results show that our SE-ORNet trained on the synthetic SURREAL dataset still produces impressive performance on the real scanned dataset, demonstrating strong generalization and robustness.

5. Conclusion

In this paper, we propose a self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. To the best of our knowledge, SE-ORNet is the first self-ensembling network in this area. To solve the mismatching issue of symmetrical parts in point clouds with different body orientations, we design a plug-and-play lightweight orientation estimation module to align the orientations of two point clouds. Extensive experiments conducted on four shape correspondence benchmarks demonstrate the superior performance of SE-ORNet.

6. Acknowledgements

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078, Grant 12150007) and National Defense Basic Scientific Research program of China (Grant JCKY2020903B002).

References

[1] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006. 1, 2

[2] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 6

[3] Benedict J Brown and Szymon Rusinkiewicz. Global non-rigid alignment of 3-d scans. In *ACM SIGGRAPH 2007 papers*, pages 21–es. 2007. 1

[4] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 6, 7

[5] Meng Ding and Guoliang Fan. Articulated and generalized gaussian kernel correlation for human pose estimation. *IEEE Transactions on Image Processing*, 25(2):776–789, 2015. 1

[6] Roberto M Dyke, Yu-Kun Lai, Paul L Rosin, Stefano Zappala, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020. 7

[7] S. et al. Implicit field supervision for robust non-rigid shape matching. In *ECCV*, 2022. 7

[8] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 1, 2, 5, 6, 7

[9] Qi-Xing Huang, Bart Adams, Martin Wicke, and Leonidas J Guibas. Non-rigid registration under isometric deformations. In *Computer Graphics Forum*, volume 27, pages 1449–1457. Wiley Online Library, 2008. 1, 2

[10] 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. 7

[11] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 3

[12] 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. 1, 2, 3, 5, 6, 7

[13] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 2

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 4

[15] Roee Litman and Alexander M Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013. 1, 2

[16] 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. 1, 2, 6

[17] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, volume 7, page 3, 2019. 5

[18] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 4

[19] 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):1–11, 2012. 1, 2

[20] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 4

[21] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 2

[23] Haoxi Ran, Wei Zhuo, Jun Liu, and Li Lu. Learning inner-group relations on point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15477–15487, 2021. 2

[24] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015. 3

[25] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37(6):1–16, 2018. 7

[26] Yang-Tian Sun, Qian-Cheng Fu, Yue-Ren Jiang, Zitao Liu, Yu-Kun Lai, Hongbo Fu, and Lin Gao. Human motion transfer with 3d constraints and detail enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[27] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3, 4, 5

[28] Art Tevs, Alexander Berner, Michael Wand, Ivo Ihrke, and H-P Seidel. Intrinsic shape matching by planned landmark

sampling. In *Computer graphics forum*, volume 30, pages 543–552. Wiley Online Library, 2011. 1, 2

[29] 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)*, 38(5):1–12, 2019. 1, 2, 3, 5, 6

[30] Yi Xu, Yao Lu, and Ziyu Wen. Owlii dynamic human mesh sequence dataset. In *ISO/IEC JTC1/SC29/WG11 m41658, 120th MPEG Meeting*, volume 1, 2017. 8

[31] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5589–5598, 2020. 2

[32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6052–6061, 2021. 1, 2, 3, 6, 7

[33] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2

[34] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4965–4974, 2021. 2

[35] 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. 6