2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

3DMotion-Net: Learning Continuous Flow Function for 3D Motion Prediction

Shuaihang Yuan*, Xiang Li*, Anthony Tzes, Yi Fang†
NYU Multimedia and Visual Computing Lab, USA
New York University Abu Dhabi, UAE
New York University, USA

***Abstract*— This paper deals with predicting future 3D motions of 3D object scans from the previous two consecutive frames. Previous methods mostly focus on sparse motion prediction in the form of skeletons. While in this paper, we focus on predicting dense 3D motions in the form of 3D point clouds. To approach this problem, we propose a self-supervised approach that leverages the power of the deep neural network to learn a continuous flow function of 3D point clouds that can predict temporally consistent future motions and naturally bring out the correspondences among consecutive point clouds at the same time. More specifically, in our approach, to eliminate the unsolved and challenging process of defining a discrete point convolution on 3D point cloud sequences to encode spatial and temporal information, we introduce a learnable latent code to represent the temporal-aware shape descriptor, which is optimized during the model training. Moreover, a temporally consistent motion Morpher is proposed to learn a continuous flow field which deforms a 3D scan from the current frame to the next frame. We perform extensive experiments on D-FAUST, SCAPE, and TOSCA benchmark data sets. The results demonstrate that our approach is capable of handling temporally inconsistent input and produces consistent future 3D motion while requiring no ground truth supervision.**

I. INTRODUCTION

3D motion prediction has been a long-standing problem in computer vision and robotics, serving as an essential building block of many applications, such as object tracking, 3D rigging, virtual and augmented reality, human-robot interaction, and motion synthesis [1], [2], [3]. However, this task is still challenging due to sensor noise, occlusion, and dynamic topology changes, especially for real-world scans, which exhibit significant shape deformations and variations.

Existing motion prediction works mostly focus on predicting future object poses characterized by skeletons from a given past. Early efforts build their model with expert knowledge about motion patterns in the form of Markovian assumptions [4], smoothness, or low dimensional embeddings. Recent works use deep recurrent neural networks (RNNs) to better capture temporal characteristics [5]. Another family of methods leverages the power of generative adversarial networks (GAN) to enforce the predicted skeletons to be similar to a real sequence of poses [6].

* Those authors contributed equally to this paper. † Yi Fang is the corresponding author.

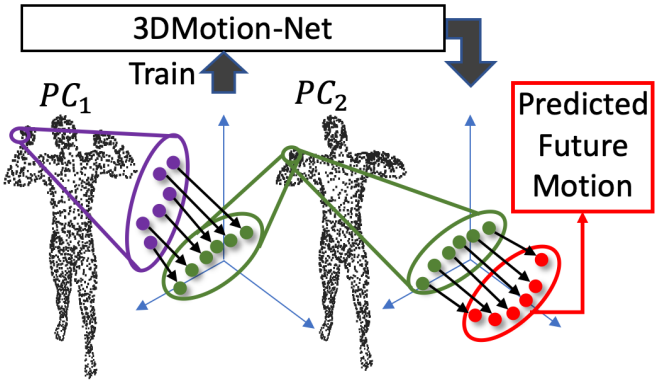

Fig. 1. Illustration of 3D Motion prediction. Our model takes two consecutive point sets in 3D space as input and predicts the flow between two consecutive point sets. The learned motion patterns are used to infer future motion by predicting the flow field from the current frame to the future one.

In contrast to previous researches, we aim at predicting dense 3D motions in the form of 3D point clouds rather than skeletons. Unlike skeletons that have pre-defined structures and with fewer node points, 3D point clouds are unstructured and unordered, which makes this task even more challenging. There are two main challenges for the task of dense 3D motions prediction from point clouds: 1) first, it's still a difficult task to learn robust and representative feature embeddings from 3D point clouds; 2) second, it's still unknown how to model the temporal correlation among consecutive frames. For the first challenge, researches have proposed extensive hand-crafted signatures [7], [8], [9], [10] to extract point features from 3D point clouds. Recent researches leverage the power of deep neural networks to encode the geometric information of input shapes to deep feature descriptors [11]. Following the pioneering work of PointNet [12] that directly applies convolution on raw point clouds, various point convolutions operations have been developed to learn shape descriptors from 3D scans [13], [14]. Nevertheless, it's still challenging and non-trivial to find the best convolution operation that is powerful and robust enough to deal with real-world scans.

Regarding the second challenge of learning spatiotemporal correlations among consecutive frames, this area has less research attention. Previous learning-based methods mostly

focus on modeling 3D motions of existing observations and can not deal with the problem of predicting future motions from previous observations. In addition to the spatial geometric features, the temporal features also play an important role in predicting the motions for time-varying 3D scans. However, it would be a challenging problem to define a convolution operation that is powerful enough to extract robust spatial and temporal features from time-varying scans. In [15], the authors propose probably the first spatiotemporal representation of time-varying 3D point clouds and verify the effectiveness for dense 4D reconstruction. Nevertheless, they still use PointNet-like encoder network for point feature learning of a sequence of point clouds.

In this paper, to avoid the challenging issue of spatiotemporal feature extraction using an encoder network, we introduce a learnable latent code to represent the temporal-aware shape features which can be optimized during model training. This temporal-aware shape representation is in nature captures both spatial geometry characteristics of 3D shapes and their correlation among consecutive frames. Our method is self-supervised, assuaging the supervision of ground-truth correspondences and predicting future motion and bringing out correspondences of future motion naturally. The pipeline of our proposed method is presented in Fig.2. Three major models are enclosed to learn the desired 3d motion flow function. We first propose a temporal-aware shape descriptor optimizer, shown in Fig.3 that can learn the desired spatiotemporal representations implicitly. A temporally consistent motion, Morpher, which takes as input a sequence of two 3D source shapes, is then used to deform the source shapes to the corresponding target shapes based on temporal-aware shape descriptor. This morphing module is built upon a deep neural network. Both temporal-aware shape descriptors and the morphing decoder network are optimized during the network training process using an unsupervised loss function. While in the test stage, the decoder network parameters are fixed, which can be regarded as learned prior information from the training data set, and we only optimize the temporal-aware shape descriptor towards the optimal spatiotemporal representation. The future motion with dense correspondence can be generated using the optimal spatiotemporal representation.

In summary, our main contributions are:

- We present a self-supervised method that can learn continuous flow function from a sequence of 3D scans. This function can predict the future object motions and naturally bring out the shape correspondences simultaneously.
- We introduce a novel spatiotemporal feature representation for time-varying 3D scans, which eliminates the unsolved and challenging process of defining a discrete point convolution on continuous space to encode spatial and temporal information.
- We conduct extensive experiments on several public data sets and demonstrate that our proposed method can be applied to temporally inconsistent scans to predict a sequence of temporally consistent motions.

II. Related Works

In this section, we first go through some feature learning techniques for 3D representations. Then, we summarize existing flow estimation methods and temporal feature learning approach.

A. 3D Feature Learning

Though deep learning has achieved significant progress in various image recognition tasks, the research on applying deep learning on raw point cloud to obtain a robust and informative feature is still challenging. There are multiple ways to represent 3D data, such as raw point clouds, voxels, meshes, leading researchers to different ways towards 3D signature learning. Voxel Grid is a popular and adaptive data structure in 3D, which can be easily applied to 3DConvNet [16] to obtain voxel-wise features. The exceptional work of Wu [16] firstly brings the success of CNN from 2D images into 3D. With a high-resolution volumetric representation of 3D Shape, most of the position and geometric information can be well preserved, making it possible to actively learn shape descriptors from the voxel-based representations. However, a high resolution of a voxel grid usually leads to cubically growing of space and computational cost. Some recent jobs focus on applying deep learning on meshes. These methods, nevertheless efficient in manifold meshes, are hard to obtain a generalized feature that robust to structure variations. Multiview-based methods, e.g., [17], transform 3D shapes into multiple 2D images and aggregates features of each 2D image to specialize in the feature of a 3D shape. Later work on multiview [18], extends this idea to multiple 3D tasks and show very persuasive performance. Even though those multiple views are very applicable and robust in some specified tasks, they are sensitive to 3D data resolution, and they can only get information from the object surface. Among all 3D data types, point cloud, which is capable of rendering meticulous 3D models, is the most informative and rawest one. The information around a specified point and the information of the general object are usually refined as local features and global feature. Due to its irregularity, most feature extraction approaches with point cloud are hand-crafted for a specific task. In [12], the authors first introduce the network that directly consumes raw point clouds. PointNet uses a max-pooling layer to achieve invariant to different permutations of the point cloud. Extensive following works [13], [19] developed various point convolution operators to directly learn point signature from point clouds. Notable method PointNet++ [13] groups points in different scales to extract the features hierarchically to obtain multi-scale point descriptors. Although these methods have achieved state-of-the-art performance on various point cloud processing tasks, they usually need to carefully design the point convolution operators and network architectures to ensure the quality of point descriptors. In contrast, DeepSDF [20] proposes an auto-decoder architecture that abandons the point feature encoding network and only use a decoder network to achieve 3D shape completion. A learnable latent code is designed

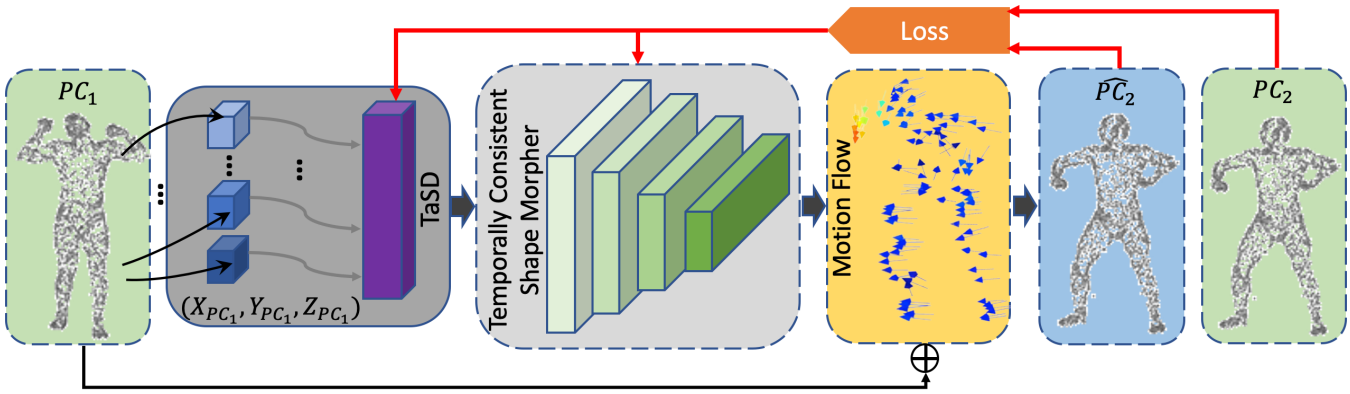

Fig. 2. This is the pipeline of the proposed method. We proposed two models to learn a continuous flow function that can predict future motion. The first one is a temporally-aware shape descriptor that encoders the essence of spatiotemporal motion patterns between consecutive frames, shown in the purple box. The second module is a temporally consistent shape Morpher, that generates a 3D flow field from the current frame to the next frame, represented in the red block. Our module takes two consecutive point cloud PC_1 and PC_2 as inputs to train the Morpher and optimize the TaSD. We concatenate the TaSD to each point of PC_1 and feed it to the shape Morpher to obtain the predicted 3D motion. After applying the motion flow to PC_1 we get the predicted point cloud $\hat{PC}_2$, which is compared with PC_2 to formulate the loss term. During training, we optimize both shape Morpher network and the latent code z. While in the inference state, we fix the shape Morpher network and only optimize latent code z towards the minimization of the loss between $\hat{PC}_2$ and PC_2 and then predict the future frame $\hat{PC}_3$.

to characterize the global feature representation, which is randomly initialized and optimized during network training.

B. 3D Flow Estimation

It is of great importance to study the problem of flow estimation on 3D point clouds in the robotics field. Many flow estimation of 3D point clouds using classical techniques has been well studied such as [21] which apply the occupancy maps to scenes and use particle filters or EM to estimate the motion flow of the object. Besides the classical techniques, Dewan et al. [22] proposed using the hand-crafted descriptor with a regulation term to estimate the rigid motion flow of a scan. In addition to the hand-crafted descriptors, researchers leverage the power of deep neural networks to learn the descriptor. PointFlowNet builds on the VoxelNet [23] to predict scene flow and rigid object motion. FlowNet3D [24] uses the PointNet++ [13] as the backbone to embed the flow. Instead of using PointNet++, HPLFlowNet [25] use Bilateral Convolutional Layer (BCL) for a better global information extraction. Though previous works achieve a promising result for flow estimation, these methods focus on the finding explicit design of convolutional architectures to learn a better shape descriptor and fail to explore the temporal information between a sequence of inputs. In this work, we introduce a novel spatiotemporal feature representation for time-varying 3D scans which eliminates the unsolved and challenging process of defining a discrete point convolution on continuous space to encode spatial and temporal information.

III. Problem Statement

We leverage the power of deep neural networks to learn a continuous motion flow function for 3D motion prediction from consecutive frames of point clouds. Our model takes as input two sets of points at consecutive frames, which are randomly sampled from a time-varying 3D objects. Each point in the point sets has 3 dimensions corresponding to the (X,Y,Z) coordinates in the Euclidean space. It is flexible to use additional features such as point colors as the extra inputs but in this paper, we focus on the 3 dimensions inputs only.

Based on the settings above, we sample a set of points P from one frame. For the next consecutive frame, we sample another set of points Q. Note that due to background noises, occlusion and dynamic topology changes, P, and Q are not necessarily have corresponding relationships. Now, let us suppose a point (x_i, y_i, z_i) is the ith point out of N sampled points in the first frame, its corresponding location at the second frame Q becomes (x_i', y_i', z_i'). For this i^th point at the first frame, the motion is represented by a point flow f_i, such that $f_i = (x_i', y_i', z_i') - (x_i, y_i, z_i)$. Consequently, For the whole point set P, there exist a motion flow F such that F contains point flow for every point at the first frame to the next frame $\mathbf{F} = \{f_i\}_{i=1}^N$. This paper aims to predict the motion flow from the second frame Q to the future frame Y driven by the existing motion flow from P to Q.

IV. 3DMotion-Net Architecture

In this section, we introduce our motion flow prediction network, 3DMotion-Net, which alleviate the explicit design of encoder to extract spatial and temporal feature, see 2 for illustration. Our proposed method consists of two key components. The first component is a temporal-aware shape descriptor optimizer that tries to learn spatiotemporal representations. The second component is temporally consistent motion Morpher, which learns a continuous motion flow function that morphs a 3D scan from current frame to the next frame. We first present these two modules and then introduce the detailed network architecture.

A. Shape Descriptor Learning with Temporal-Aware Shape Descriptor Optimizer

Many previous works leverage the neural network to design a task-specific architecture for shape descriptor extraction. To apply discrete convolution to continuous 3D space especially to irregular and unordered point cloud, numerous point-based convolution operations, such as PointNet [12], are proposed for a better point feature extraction. However, it is unclear whether there exists a best explicit feature extractor or it is necessary to use a complex extractor for feature extraction. This motivates us to reduce the dependency on the explicit design of feature extractor. We develop a novel learnable descriptor in latent space that requires no explicit feature extraction networks.

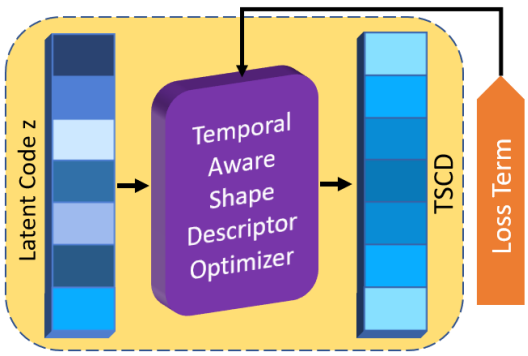

Fig. 3. Illustration of TaSD Optimizer which learn the descriptor by optimizing a latent code z using pre-defined loss.

As we showed in Fig.2, for every input pair of point sets (P,Q), we assume that there exists a latent code z_{PQ} that carries the geometric essences of two consecutive shapes as well as the translational relationship. To find the desired latent code for each input pair, we first randomly initialized this latent code from a Gaussian normal distribution, i.e., $z \sim \mathcal{N}(0,1)$. Then, the temporal-aware shape descriptor (TaSD) optimizer is applied to the initialized latent code to search for the optimum representation. We adopt the back-propagation strategy along with an unsupervised loss to optimize latent code z to obtain the optimal descriptor. The loss function and the optimization rule of the z are presented in the sections IV-C.

B. Motion Flow Predicting with Temporally Consistent Shape Morpher

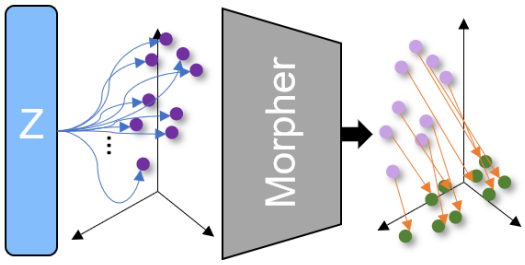

Fig. 4. The Illustration of temporally consistent shape Morpher that is used to decode optimized TaSD to continuous flow field

To regress the flow from one frame to the next, we implement a deep neural network to learn the motion flow function $\mathscr{F}(\cdot)$. Given a pair of point sets (P,Q) at two consecutive frames and the corresponding latent code z_{PQ}, our inputs are formulated as follows,

$$X_{PQ} = (P, z_{PQ}) \tag{1}$$

. Then, we feed X_{PQ} to the motion flow function to obtain motion flows as we presented in Fig.4.

We use a Multilayer Perceptrons (MLPs) with softplus activation function to approximate $\mathscr{F}(\cdot)$ using θ parameters. Hence our continuous motion flow function can be approximated as follows:

$$\mathscr{F}(X_{PQ}) \approx MLP_\theta([P, z_{PQ}]) \tag{2}$$

, where $[\cdot,\cdot]$ represent the concatenation operation. The motion flows can be obtained by the following equation:

$$F_{PQ} = \mathscr{F}(X_{PQ}) \tag{3}$$

, where F_{PQ} is the predicted motion flows from P to Q. Ideally, the predicted motion flow should be accurate enough such that point set $P + F_{PQ}$ is close to Q.

C. Motion Flow Loss and Optimization

To ensure a good quality of predicted motion flow, we design a loss term $\mathscr{L}_{\theta,z}$, which is the summation of losses of two flows. For each flow, the loss is calculated by using the Chamfer distance:

$$\mathscr{L}_{\theta,z_{PQ}}(P+F_{PQ},Q) = \frac{1}{|P+F_{PQ}|}\sum_{x\in P+F_{PQ}} \min_{y\in Q} \|x-y\|_2 + \frac{1}{|Q|}\sum_{x\in Q} \min_{y\in P+F_{PQ}} \|x-y\|_2 \tag{4}$$

. Note that Chamfer distance is calculated without using the ground truth correspondence information. This loss works as a key component in both training and inference state.

During training, our goal is to find the optimal network parameters θ and latent code z_{PQ} such that:

$$\arg\min_{\theta, z_{PQ}} \mathscr{L}_{\theta,z_{PQ}}(P+F_{PQ},Q) \tag{5}$$

Note that both network parameters θ and latent code z are updated simultaneously by this loss term during the training time.

In the inference state, we fix decoder parameters θ and reinitialize the latent code z for each input pair. The learned network parameters θ can be regarded as prior information to guide the optimization process of latent code. During the test optimization process, since we do not know the future frame Y during the inference, we use the loss term $\mathscr{L}_{\theta,z_{PQ}}(P+F_{PQ},Q)$ to optimize our latent code z using the updating rule:

$$z_{\hat{PQ}} = \arg\min_{z_{PQ}}(\mathscr{L}_{\theta,z_{PQ}}(P+F_{PQ},Q)) \tag{6}$$

Once the loss converges, the prediction of a future motion can be inferred by applying the predicted motion flow to the second frame. Moreover, the correspondence can be obtained naturally after applying the motion flow to the original point set.

V. Experiment

A. Data Sets, Implementation and Training Detail:

We conduct experiments on several public data sets to show the abilities of our model in predicting 3D motions. Experimental data sets include Dynamic FAUST (D-FAUST), a dynamic human motion data set, which contains 40,000 meshes, a human pose data set SCAPE [26] which consists of a sequence of real human scans. The TOSCA [27] data set contains synthetic nonrigid models of 80 objects with various object category, such as cat, dog, wolf, etc. Note that although the ground truth corresponding points are provided for all these three data sets, our proposed method uses no ground truth information neither in the training nor optimization.

The pipeline of our proposed method can be found in Fig.2. We generate training and testing episodes from the training and test set, respectively, each with 3 frames. For each frame, we randomly initialize a latent vector from a normal distribution, i.e.,$z \sim \mathcal{N}(0,1)$. It is possible that we can add more frames for training and testing and the performance may increase as we increase the observed frame. However, this will cause significantly increase in computational resources. By default, we set the dimension of z to 256. The temporally consistent shape Morpher is implemented using an MLP with hidden layers of size 512, 256, 128, 64, and 3 to predict a flow field from the current frame to the next frame, and the shape correspondences can be naturally generated by using this flow function.

B. Evaluation Metrics:

We use the cumulative matching accuracy to evaluate the quality of the predicted future motion. For each point, it is considered as a correct match if the distance between this point and its corresponding ground-truth point is within a certain threshold of Δ. In addition to the cumulative matching accuracy, we also use chamfer distance and ℓ_2-distance to evaluate the quality of prediction.

C. Motion Prediction on Consistent data set

Experiment Settings: To verify the effectiveness of our proposed method, we first experiment on the temporally consistent data on the SCAPE and D-Faust data set. We follow the training and testing procedure provided in the method part. We tested the performance of our model with different sizes of latent code z. Experimental results show that different sizes of z have a small effect on the final performance. We set the dimension of z to 256 in the following sections.

Result Analysis: The visualized result on consistent inputs can be found in Fig.5. The qualitative of our approach on both SCAPE and D-Faust dataset can be found in Table.I and the blue line in the Fig.7. As shown in this figure, our method can successfully predict the 3D motion from one frame to the next with satisfying correspondences.

Notice that it is not practical and unfair to compare with other related methods [15], [28] due to the following reasons. First, the existing related models take all frames as input to embed the temporal relationship. In contrast, in our model, the temporal relationships are explicitly embedded by propagating the TCD of the previous frame to the next frame to capture the locally integrated features (i.e., the feature of each frame). Second, [15], [28] exploring the temporal information are not designed to address 3D motion prediction. Third, our model is designed to address the critical problem of predicting the non-rigid future motion, which is much more complicated to predict the rigid transformation.

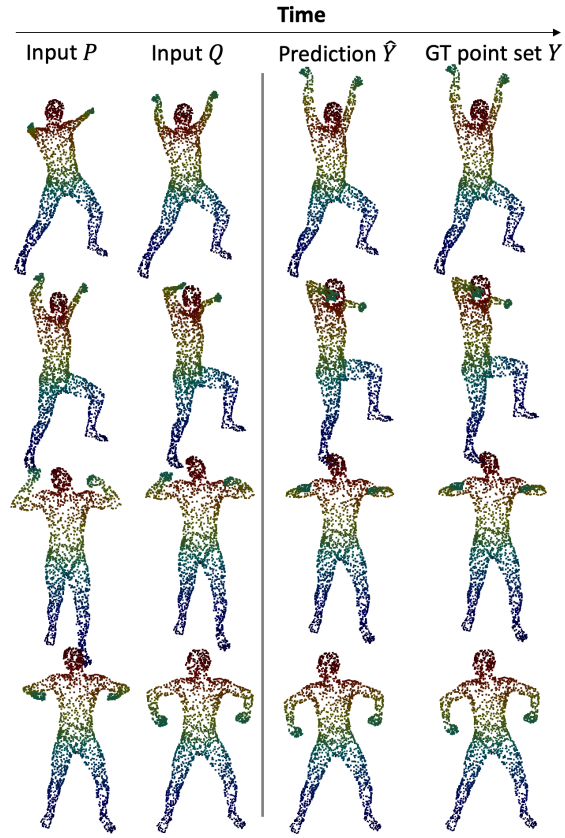

Fig. 5. Qualitative results on the temporal consistent SCAPE data set. Corresponding points are painted with same color.

3DMotion-Net		
Data Set	Chamfer Distance	Correspond. Distance
SCAPE	0.021	0.088
D-FAUST	0.029	0.091

TABLE I

Quantitative results on D-Faust dataset. Chamfer distance and correspondence ℓ_2-distance are reported. (Lower is better)

D. Motion Prediction on Inconsistent data set

Experiment Settings:

3D scans are usually captured with data noise, partial observation, and holes. To show that our proposed can handle temporally inconsistent inputs, we first conduct experiments to test the noisy resistance ability by predicting future

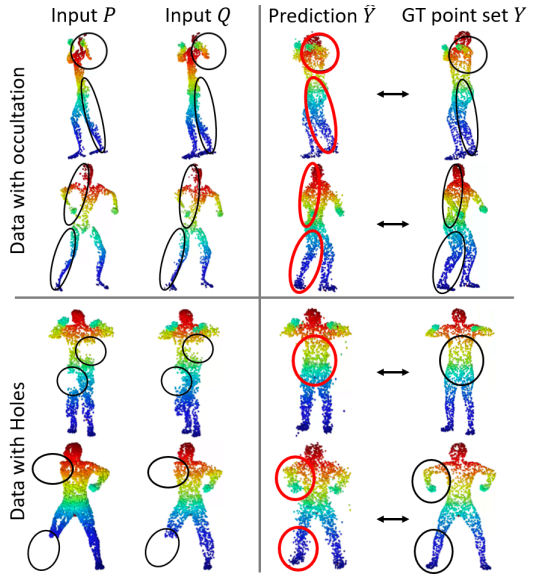

Fig. 6. Qualitative results on the inconsistent data with noises and occlusions. The high light part in this figure clearly shows that our proposed method can produce consistent prediction even when the inconsistent inputs are given.

motion on the noisy motions at the different noise levels. Therefore, we add Gaussian noise to the SCAPE data set with noise levels of 0.02 and 0.04. Moreover, to further show our model's robustness, we experiment with different input corruptions. We randomly cut off some parts of the object in the SCAPE data set to create the partial data set.

3DMotion-Net		
Data Set	Chamfer Distance	Correspond.
SCAPE w/ Nosie Level 0.2	0.029	0.091
SCAPE w/ Nosie Level 0.4	0.037	0.098
SCAPE w/ Holes	0.031	0.092
SCAPE w/ Partial	0.034	0.094

TABLE II

QUANTITATIVE RESULTS ON THE INCONSISTENT DATA. CHAMFER DISTANCE AND CORRESPONDENCE ℓ_2-DISTANCE ARE REPORTED. (LOWER IS BETTER)

Result Analysis: Our model is trained on the temporally consistent data during training process and learns the prior information of how to decode latent codes into spatial consistent flow fields. This ensures our model can still search the optimal latent codes for motion prediction with inconsistent input data. From the Fig.6, one can see that our method achieves a reasonable performance even with input noises and missing parts. This demonstrate that even when input data is inconsistent, our method can still use the designed loss to search the optimal latent code z and to well predict the future motion. This can still be proved by the quantitative results in Table.II

E. Cross-category Motion Prediction

Experiment Settings:

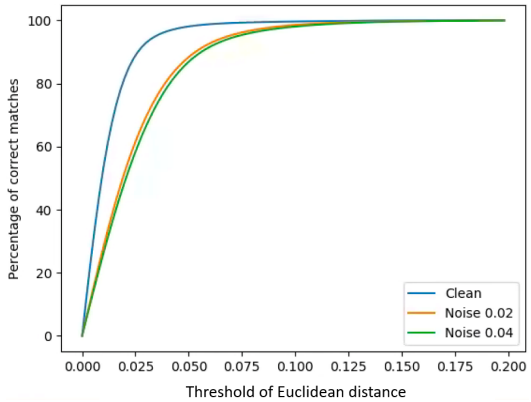

Fig. 7. Cumulative matching accuracy of our approach when taking different noise level of SCAPE data as inputs

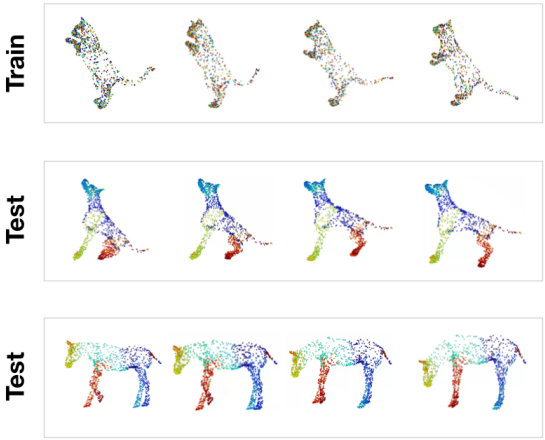

Fig. 8. Cross-category Motion Prediction. Our model is trained on cat category and evaluated on dog and horse categories. Corresponding points are painted with same color.

As described in the method section, our Temporal-aware shape descriptor (TaSD) is designed to learn temporal flow patterns between the current frame and the next frame regardless of the object categories. To verify the class transferring abilities of our proposed TaSD, we carry out further experiments on the TOSCA data set that we train our model on the cat category, and evaluate the motion prediction performance on other two categories, including dog and horse.

Result Analysis: As presented in the Figure.8,all corresponding points are consistent across all the frames. As shown in Table.III, our model gets similar performance for horse and dog categories compared to the cat category. This demonstrates that our proposed TaSD is powerful to extract temporal flow patterns regardless the object category. Qualitative results in Figure.8 also proves the same finding.

VI. CONCLUSION

This paper introduces a self-supervised method that can learn continuous flow function from two consecutive 3D scans. This function can predict the future object motion and

3DMotion-Net		
Inference Category	Chamfer Distance	Correspond.
Cat (Training)	0.020	0.087
Dog (testing)	0.023	0.090
Horse (testing)	0.023	0.089

TABLE III

QUANTITATIVE RESULTS OF THE CROSS-CATEGORY EXPERIMENT. OUR MODEL IS TRAINED ON TOSCA "CAT" CATEGORY AND THE CHAMFER DISTANCE AND CORRESPONDENCE ℓ_2-DISTANCE ARE REPORTED. (LOWER IS BETTER)

naturally bring out the shape correspondences simultaneously using the object prior and observed motion flow provided by the input two consecutive 3D scans. To eliminate the unsolved and challenging problem of defining a discrete point convolution on continuous space to encode spatial and temporal information, we introduce a temporal-aware motion descriptor to represent temporal and geometric information. A shape morph decoder network is designed to learn a continuous motion flow function which morphs a 3D scan from the current frame to the next frame. A sequence of time-varying 3D scans is used to jointly train the decoder and temporally aware motion descriptor without the supervision of ground truth. Experiments on several benchmark data sets demonstrate our model's satisfying motion prediction performance on both clean and inconsistent inputs.

REFERENCES

[1] D. Van Krevelen and R. Poelman, "Augmented reality: Technologies, applications, and limitations," *Vrije Univ. Amsterdam, Dep. Comput. Sci*, 2007.

[2] F. Zhou, H. B.-L. Duh, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ismar," in *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*. IEEE Computer Society, 2008, pp. 193–202.

[3] P. Gururajan, M. Gandhi, J. Jackson, and A. Levinshtein, "Gaming object position analysis and tracking," Nov. 9 2006, uS Patent App. 11/384,429.

[4] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "A non-parametric bayesian network prior of human pose," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1281–1288.

[5] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2891–2900.

[6] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1418–1427.

[7] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *arXiv preprint arXiv:1905.06113*, 2019.

[8] C.-H. Huang, B. Allain, J.-S. Franco, N. Navab, S. Ilic, and E. Boyer, "Volumetric 3d tracking by detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3862–3870.

[9] E. Kruse and F. M. Wahl, "Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 1, May 1998, pp. 662–667 vol.1.

[10] T. P. Kucner, M. Magnusson, E. Schaffernicht, V. H. Bennetts, and A. J. Lilienthal, "Enabling flow awareness for mobile robots in partially observable environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1093–1100, 2017.

[11] Y. Fang, J. Xie, G. Dai, M. Wang, F. Zhu, T. Xu, and E. Wong, "3d deep shape descriptor," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2319–2328.

[12] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.

[14] N. Verma, E. Boyer, and J. Verbeek, "Feastnet: Feature-steered graph convolutions for 3d shape analysis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2598–2606.

[15] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4d reconstruction by learning particle dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5379–5389.

[16] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes," 2014.

[17] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," Tech. Rep. [Online]. Available: http://vis-www.cs.umass.edu/mvcnn.

[18] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, and H. Huang, "Projective Feature Learning for 3D Shapes with Multi-View Depth Images," in *Computer Graphics Forum*, 2015.

[19] L. Wang, J. Chen, X. Li, and Y. Fang, "Non-rigid point set registration networks," *arXiv preprint arXiv:1904.01428*, 2019.

[20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[21] A. K. Ushani, R. W. Wolcott, J. M. Walls, and R. M. Eustice, "A learning approach for real-time temporal scene flow estimation from lidar data," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5666–5673.

[22] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Rigid scene flow for 3d lidar scans," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1765–1770.

[23] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.

[24] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 529–537.

[25] X. Gu, Y. Wang, C. Wu, Y. J. Lee, and P. Wang, "Hplflownet: Hierarchical permutohedral lattice flownet for scene flow estimation on large-scale point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3254–3263.

[26] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "Scape: Shape completion and animation of people," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 408–416, July 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073207

[27] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[28] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.