RFNet-4D: Joint Object Reconstruction and Flow Estimation from 4D Point Clouds

Tuan-Anh Vu[1], Duc Thanh Nguyen[2], Binh-Son Hua[3], Quang-Hieu Pham[4], and Sai-Kit Yeung[1]

[1] Hong Kong University of Science and Technology
[2] Deakin University
[3] VinAI Research
[4] Woven Planet North America, Level 5

Abstract. Object reconstruction from 3D point clouds has achieved impressive progress in the computer vision and computer graphics research field. However, reconstruction from time-varying point clouds (a.k.a. 4D point clouds) is generally overlooked. In this paper, we propose a new network architecture, namely RFNet-4D, that jointly reconstruct objects and their motion flows from 4D point clouds. The key insight is that simultaneously performing both tasks via learning spatial and temporal features from a sequence of point clouds can leverage individual tasks, leading to improved overall performance. To prove this ability, we design a temporal vector field learning module using unsupervised learning approach for flow estimation, leveraged by supervised learning of spatial structures for object reconstruction. Extensive experiments and analyses on benchmark dataset validated the effectiveness and efficiency of our method. As shown in experimental results, our method achieves state-of-the-art performance on both flow estimation and object reconstruction while performing much faster than existing methods in both training and inference. Our code and data are available at https://github.com/hkust-vgd/RFNet-4D.

Keywords: dynamic point clouds, 4D reconstruction, flow estimation

1 Introduction

Literature has shown several breakthroughs in deep learning for reconstruction of 3D models from point clouds. Recently, the research community has seen great successes in neural representations using implicit fields [27,32,5,28], which pave an effective way on how 3D data can be represented by neural networks. Unlike traditional representations that are often realised in discrete forms (e.g., discrete grids of pixels in image representation, discrete grids of voxels in 3D object representation), the neural implicit representation parameterises a signal as a continuous function via a neural network. This function maps a signal from its original domain, which can be queried at any resolution, to an output domain that captures some properties of the query. Most existing methods focus

on neural representation of 3D data in static conditions. However, in reality, real-world objects exist in a dynamic environment that changes over time and space, and thus cannot be well modelled using implicit representations applied to static shapes. Approaches for 4D reconstruction (i.e., reconstruction of a 3D object over time) have been explored but they often need expensive multi-view settings [22,7,29,30]. These settings rely on a template model (of the object) with fixed topology [1,18,40,47], or require smooth spatio-temporal input [33,42], and thus limiting their applicability in practice.

To enable object reconstruction directly from 4D data without predefined templates, OFlow [31], a pioneering method for 4D reconstruction, was developed to calculate motion fields of 3D points in a 3D point cloud in space and time to implicitly represent trajectories of all the points in dense correspondences between occupancy fields. To learn the motion fields in both space and time domain, OFlow made use of a spatial encoder to learn the spatial structure of the input point cloud and a temporal encoder to learn the temporal changes of the point cloud in time. Despite impressive reconstruction results, this paradigm has a number of drawbacks. First, its spatial encoder does not take geometric attributes from numerous frames into consideration, impairing the capacity to precisely reconstruct geometric surfaces. Neither does its temporal encoder take into account temporal correspondences, which are critical for accurately capturing temporal dynamics. Second, errors in prediction of temporal continuity and reconstructed geometries are accumulated by integral of estimated instantaneous findings. Third, OFlow is trained using supervised learning. This requires correspondence labelling for all 3D points across frames in training data, leading to high labelling cost and low scalability. Fourth, the method exhibits low computational efficiency in both training and inference phase. This is due to expensive computations required to sequentially determine trajectories of 3D points throughout time by solving complex ordinary differential equations.

To address the aforementioned challenges, we propose a network architecture, namely RFNet-4D, for 4D reconstruction and flow estimation of dynamic point clouds. Our key idea is to jointly perform two tasks: 4D reconstruction and flow estimation with an intention that each task can leverage the other one to improve the overall performance. Specifically, our network takes a sequence of 3D point clouds of an object over time as input, then encodes the point clouds into spatio-temporal representations using a compositional encoder. These spatio-temporal representations are formed inclusively. In particular, the spatio-temporal representation of a point cloud at a time step is calculated from the spatial layout of points in that point cloud and the temporal changes of the points in the point cloud throughout time. The spatio-temporal representations are then decoded by a joint decoder which jointly reconstructs the object and predicts a motion vector for each point in the reconstructed object throughout time. The entire network can be trained end-to-end, where the reconstruction and flow estimation task are trained with supervised and unsupervised learning, respectively. Our method also allows fast computations of spatial and temporal features as those computations can be performed in parallel. This is another advantage of our
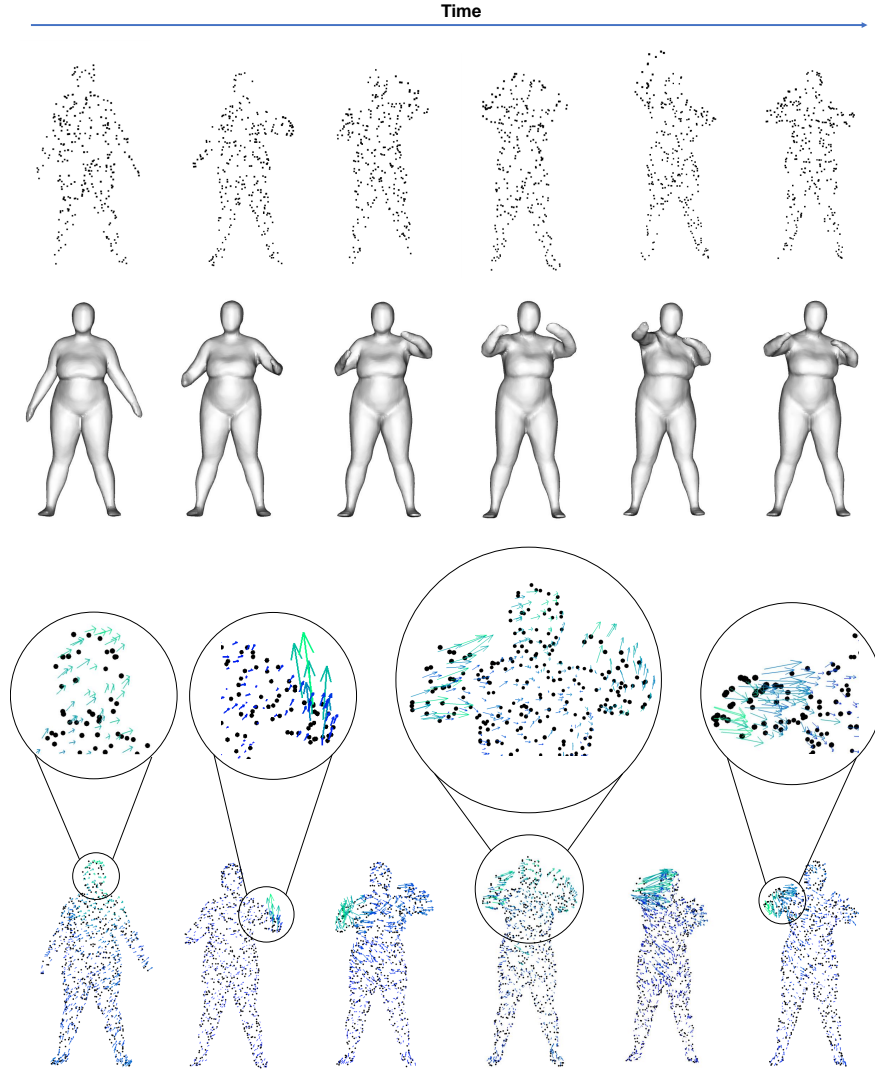
Time

Fig. 1: **Summary of our method.** Given a sequence of time-varying 3D point clouds (first row), we jointly reconstruct corresponding 3D geometric shapes (second row) and estimate their motion fields for every point cloud (third row).

method, compared with OFlow which estimates the motion flows sequentially and thus often experiences time lags. We illustrate several results of our method in Fig. 1. In summary, the contributions of our work are as follows:

- RFNet-4D: a network architecture for joint object reconstruction and flow estimation from a sequence of time-varying 3D point clouds.

- A joint learning method for training the proposed RFNet-4D using both supervised and unsupervised learning, and in both forward and backward time direction. To the best of our knowledge, this learning mechanism is novel, and its benefit is verified throughout experiments.
- Extensive experiments and analyses that prove the effectiveness and efficiency of our proposed method on two tasks: 4D reconstruction and flow estimation.

2   Related Work

3D Reconstruction. Numerous studies have been conducted with the goal of reconstructing a continuous surface from a variety of input, including RGB images [39,46,19], point clouds [20]. Thanks to advances in deep learning, recent 3D object reconstruction approaches have resulted in significant progress. Early attempts represent reconstructed objects in regular grid of 3D voxels [45,11] or point clouds [35,10]. However, those representations cannot well capture surface details and suffer from low resolutions. Alternatively, there are methods, e.g., [44,23,17] reconstructing triangular meshes (including vertices and faces) of 3D objects. In these methods, an initial template with fixed topology is employed and the reconstruction is performed using regression. For surface representation, several methods focus on learning an implicit field function that allows more variable topology in reconstructed objects [6,4,9,16].

To extend the ability of implicit functions on representations other than traditional forms (i.e., voxels, points, meshes), occupancy maps [27,34] and distance fields [32,4] are proposed. An occupancy map of a 3D point cloud contains indicators that indicate being foreground of points in the 3D space. A distance field provides the distance from every point to its nearest surface. Since the implicit function models objects in a continuous manner, more information is preserved and more complicated shapes can be well described. For instance, Occupancy Network in [27] described a 3D object using continuous indicator functions that indicate which sub-sets of 3D space the object occupies, and an iso-surface retrieved by employing the Marching Cube algorithm [25].

4D Reconstruction. Despite being less studied compared with 3D reconstruction, literature has also shown recent attentions of the research community to 4D reconstruction, i.e., reconstruction of a sequence of 3D objects from time-varying point clouds [22,29,30]. In this section, we limit our review to only learning-based 4D reconstruction methods.

A crucial component in 4D reconstruction is motion capture and modelling. Niemeyer *et al.* [31] introduced a learning-based framework that calculates the integral of a motion field specified in space and time to implicitly represent the trajectory of a 3D point to generate dense correspondences between occupancy fields. Jiang *et al.* [15] proposed a compositional representation for 4D capture, i.e., a deformable representation that encloses a 3D shape and velocity of its 3D points over time. Such representation was composed of encoder-decoder architectures. Specifically, to simulate the motion in time-varying 3D data, a neural Ordinary

Differential Equation was trained to update the starting state of motion based on a learnt motion representation, and a decoder was used to reconstruct a 3D model at each time step using a shape representation and the updated state. They also introduced an Identity Exchange Training technique to motivate their system to learn how to decouple each encoder-decoder successfully. Tang *et al.* [38] proposed a pipeline for determining the temporal evolution of the 3D shape of the human body using spatially continuous transformation functions between cross-frame occupancy fields. By explicitly learning continuous displacement motion fields from spatio-temporal shape representations, the pipeline aims to construct dense correspondences between projected occupancy fields at different time steps.

Motion Transfer. Traditional techniques for 3D pose transfer often use discrete deformation transfers. An example of mesh deformation is described in [43], where spatially adaptable instance normalisation [14] was used to modify 3D meshes. However, this method requires a dense triangular mesh of an object to be given in advance, while there is specific mechanism to depict continuous flows in both spatial and temporal domain.

3D motion transfer is another technique for creating 3D objects from a pair of source and target object sequences. It operates by causing the target object sequence to undergo the same temporal deformation in the source object sequence. This technique can be applied to model continuous transformation of an object's pose over time. For instance, OFlow [31] transmitted motion across sequences of source and target human models by applying motion field-based representations to the targets in a predetermined manner. However, since OFlow does not explicitly differentiate the representations of pose and shape, we found that it only produces reasonable motion transfer results when the identities and initial poses of both the source and target objects are similar.

Shape Correspondence Modelling. Modelling of point-to-point correspondences between two 3D shapes is a well-studied topic in computer vision and computer graphics [2]. The goal of modelling time-varying occupancy fields is strongly related to the goal of field-based deformation [26], which we have previously discussed. However, most of these works describe the motion fields only on object surfaces. To better describe the motion flow, we argue to model the correspondences between 3D shapes in the entire 3D space.

When modelling the growth of a signed distance field, Miroslava *et al.* [37] chose to implicitly provide the correspondences rather than explicitly yielding them. They optimised an energy function capturing the similarity between the Laplacian eigenfunction representations of the input and the target shape. However, we found that their method is sensitive to noise, probably due to lack of capability of providing correspondences accurately from signed distance fields. In contrast, we learn the rich correspondences between time-varying occupancy fields based on a intuitive insight, that the occupancy values of points are always invariant during the temporal evolution of the occupation field.
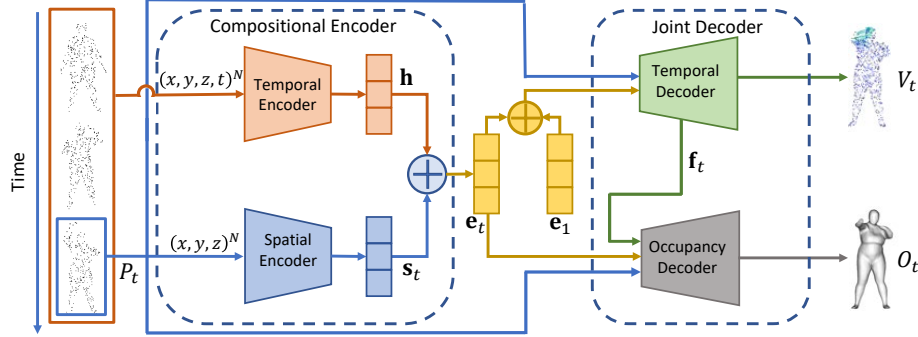
Fig. 2: **Overview of our method.** An input 3D point cloud sequence is fed into a spatio-temporal encoder to extract spatio-temporal representations. The representations are then passed via two distinct decoders, occupancy and motion decoders. In each data frame, the occupancy decoder aims to predict an occupancy field of the point cloud in the frame. Simultaneously, the motion decoder predicts the correspondences between points in the current frame and its preceding frame.

3   Our Method: RFNet-4D

3.1   Overview

Our network takes as input a sequence of sparse, incomplete, and noisy 3D point clouds $\{P_t|t = 1, ..., T\}$ where T is the length of the sequence, and each point cloud P_t is a set of 3D locations. Our aim is to simultaneously perform the following tasks:

- Reconstruct a sequence of occupancy maps $\{O_t|t = 1, ..., T\}$ where each O_t is an occupancy map of a point cloud P_t, i.e., $O_t(\mathbf{p}) = 1$ if $\mathbf{p}$ is a 3D point on the reconstructed surface of P_t, and $O_t(\mathbf{p}) = 0$, otherwise;
- Estimate a sequence of vector fields $\{V_t|t = 1, ..., T\}$ where each V_t is a 3D vector field capturing motion flows of reconstructed points of P_t, i.e., $V_t(\mathbf{p}) \in \mathbb{R}^3$ represents the motion flow of a reconstructed point $\mathbf{p}$ at time step t given a point cloud P_t.

Both tasks benefit by a compositional encoder that learns spatio-temporal representations from time-varying point clouds. The temporal features contained in these spatio-temporal representations capture holistic motion information and are computed once on the entire input point cloud sequence. This allows fast computations in following operations as spatio-temporal data can be processed at any arbitrary frame. The spatio-temporal representations are processed by a joint decoder including two decoders, each of which extracts relevant information for its downstream task (i.e., reconstruction and flow estimation). These decoders do not operate independently but co-operate closely to fulfill their tasks. To further

exploit the benefit of temporal information, we couple the reconstruction and flow estimation task in both forward and backward time direction. We present an overview of our method in Fig. 2. We describe main components of our method in the following sections.

3.2   Compositional Encoder

The compositional encoder includes a temporal encoder and a spatial encoder. There exist several manners to encode 4D point clouds. For instance, Liu *et al.* [24] applied spatio-temporal neighbourhood queries in representing 4D point clouds. However, this method requires high computational complexity. Inspired by the success and efficiency of the point cloud representation used in OFlow [31] and LPDC [38], we adopt the parallel ResNet [13] variant of PointNet [36] for both the spatial and temporal encoder (see Fig. 3). These encoders are basically similar in their architectures. The difference between them is that while the spatial encoder processes each point cloud P_t individually at a time t to generate a representation $\mathbf{s}_t$, the temporal encoder acquires the whole point cloud sequence to calculate a holistic temporal representation $\mathbf{h}$ once. These spatial and temporal representations are finally concatenated to form a spatio-temporal representation $\mathbf{e}_t$ that encodes the geometric information of a point cloud P_t in space with regard to its temporal changes (see Fig. 2). Our encoders share similar structures with the encoders in LPDC [38]. However, instead of using a complicated fusion method, we empirically found that a simple concatenation of the spatial and temporal features is good enough to effectively create spatio-temporal representations.

Since $\mathbf{h}$ is computed once on the entire input point cloud sequence, $\mathbf{e}_t$ can be extracted at any arbitrary time step t without time lags, as opposed to methods processing point clouds sequentially, e.g., OFlow [31]. Thanks to this advantage, the processing time RFNet-4D can be optimised by calculating spatio-temporal representations $\mathbf{e}_t$ in parallel.

3.3   Joint Decoder

The joint decoder takes a spatio-temporal representation $\mathbf{e}_t$ and the original point cloud sequence as input, then passes this input into two decoders (temporal decoder and occupancy decoder) to perform flow estimation and object reconstruction. Our temporal decoder and occupancy decoder are built upon the architecture from LPDC [38]. However, instead of decoupling the decoders as in [38], we hypothesized that jointly addressing two tasks by sharing information between corresponding decoders can leverage individual tasks. As a consequence, the close collaboration of flow estimation and object reconstruction allows some relaxation in the supervision need.

The temporal decoder operates as follows (see Fig. 4(a)). We first extract a spatio-temporal representation $\mathbf{e}_1$ for the first point cloud P_1 from the input sequence, using the compositional encoder. For each following point cloud P_t, we compute its spatio-temporal representation $\mathbf{e}_t$, then concatenate $\mathbf{e}_t$ with $\mathbf{e}_1$. This concatenated representation captures temporal changes of P_t in relative to P_1,
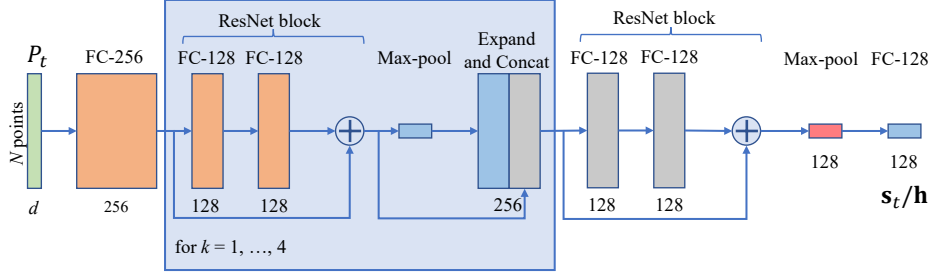
Fig. 3: **Architecture of the spatial/temporal encoder.** The input dimension d is set accordingly to a corresponding encoder. In particular, $d = 3$ (i.e., (x, y, z)-coordinates) for the spatial encoder and $d = 4$ (i.e., (x, y, z)-coordinates and time variable) for the temporal encoder. $\oplus$ indicates a concatenation operation. Output of the spatial and temporal encoder are $\mathbf{s}_t$ and $\mathbf{h}$, respectively.

and is again concatenated with all points in P_t to be processed by a series of five ResNet residual blocks [13]. Each block consists of two fully connected layers with skip connections and ReLU activation functions [12]. The outcome of these blocks is a feature map, namely $\mathbf{f}_t$. This feature map is finally passed to a fully connected layer, returning a motion field V_t describing the motion of P_t.

The occupancy decoder is slightly different from the temporal decoder (see Fig. 4(b)). Also different from all existing methods, our occupancy decoder works collaboratively with the temporal decoder. Particularly, input for the occupancy decoder to reconstruct the object at time step t includes a point cloud P_t, a spatio-temporal representation $\mathbf{e}_t$ (obtained from the compositional encoder), and a flow feature map $\mathbf{f}_t$ (returned by the temporal decoder). The point cloud P_t is first processed by a fully connected layer to extract a feature map. Similarly, the spatio-temporal representation $\mathbf{e}_t$ is fed to two different fully connected layers to obtain feature maps β and γ. These output feature maps (from P_t and $\mathbf{e}_t$) are passed to a series of five residual blocks, similar to those used in the temporal decoder. Following ONet [27], we apply Conditional Batch Normalization (CBN) introduced in [8,41] to β and γ. Finally, the flow feature map $\mathbf{f}_t$ is injected into the occupancy decoder to produce an occupancy map $O_t(\mathbf{p})$, where $O_t(\mathbf{p}) = 1$ if the point $\mathbf{p}$ belongs to the object at time step t, and $O_t(\mathbf{p}) = 0$ otherwise.

3.4   Joint Learning

Our RFNet-4D is trained by jointly performing two optimisation processes: unsupervision for flow estimation and supervision for object reconstruction. Existing works train flow estimation using supervised learning [10,27,31,38,15], requiring fully annotated point correspondences in training data. In this paper, we propose to learn point correspondences in a point cloud sequence via an unsupervised manner, thus opening ways to new applications and more training data. Specifically, let V_t be a motion field (i.e., a set of 3D vectors) at P_t, V_t is

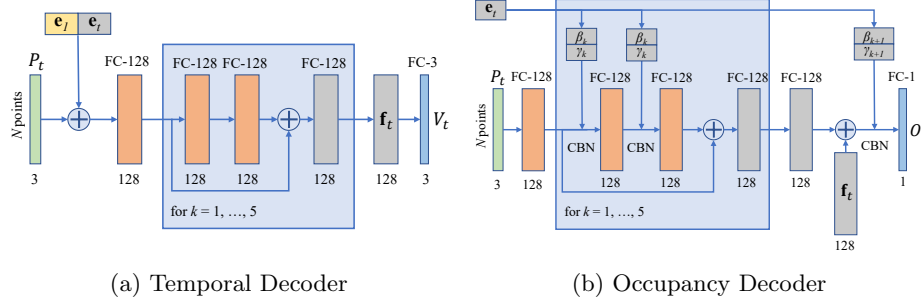(a) Temporal Decoder   (b) Occupancy Decoder

Fig. 4: **Architecture of the temporal and occupancy decoder**; $\oplus$ indicates a concatenation operation. The temporal decoder returns both a motion field V_t and a motion feature map $\mathbf{f}_t$, which is then inputted to the occupancy decoder.

estimated using the temporal decoder. We measure the correspondences between points in P_t and P_{t+1} via the Chamfer distance between P_{t+1} and a translated version of P_t made by V_t (i.e., $P_t + V_t$). We define our flow loss as follow,

$$\mathcal{L}_{flow} = \sum_t \max \left\{ \frac{1}{|P_t|} \sum_{\mathbf{p} \in P_t + V_t} \min_{\mathbf{p}' \in P_{t+1}} \|\mathbf{p} - \mathbf{p}'\|_2, \right.$$
$$\left. \frac{1}{|P_{t+1}|} \sum_{\mathbf{p}' \in P_{t+1}} \min_{\mathbf{p} \in P_t + V_t} \|\mathbf{p}' - \mathbf{p}\|_2 \right\} \tag{1}$$

Reconstruction task can be trained using supervised approach. We use conventional binary cross entropy (BCE) loss to measure the difference between predicted occupancy maps and corresponding ground truth maps. Specifically, we define our reconstruction loss as follow,

$$\mathcal{L}_{reconstruction} = \sum_t \sum_{\mathbf{p} \in P_t} \mathcal{L}_{\text{BCE}} \left(O_i \left(\mathbf{p} \right), O_i^{gt} \left(\mathbf{p} \right) \right) \tag{2}$$

where O_i^{gt} represents the ground truth occupancy map of the point cloud P_t.

Finally, we use the following loss to train the entire RFNet-4D,

$$\mathcal{L} = \mathcal{L}_{flow} + \lambda \mathcal{L}_{reconstruction} \tag{3}$$

where λ is a hyper-parameter.

To further exploit the benefit of temporal information, we train our RFNet-4D in both forward and backward direction in time. Particularly, we calculate the holistic temporal representation $\mathbf{h}$ for two sequences $\{P_1, ..., P_T\}$ (forward) and $\{P_T, ..., P_1\}$ (backward), and use $\mathbf{h}$ to encode the spatio-temporal representations $\mathbf{e}_t$ in both forward and backward time direction. As shown in our experiments, this training strategy improves the performance of our network in both object reconstruction and flow estimation task.

4   Experiments

4.1   Experimental Setup

Dataset. We trained and evaluated our method on the pre-processed data of D-FAUST dataset [3], a benchmark dataset commonly used in state-of-the-art. D-FAUST dataset contains raw-scanned and registered meshes for 129 sequences of 10 human subjects (5 females and 5 males) with various motions such as "shake hips", "running on spot", or "one leg jump". We followed the train/test split used in [31]. Specifically, we divided all the sequences in D-FAUST dataset into three sets: training set (105 sequences), validation set (6 sequences), and test set (21 sequences). Since each sequence is relatively long (with more than 1,250 time steps) and in order to increase the size of the dataset, we sub-sampled each sequence into smaller sub-sequences of 17 to 50 time steps.

Implementation Details. We implemented our method in Pytorch. We adopted Adam optimizer [21] where the learning rate γ was set to 10^{-4} and decay was set to 5,000 iterations. We empirically set λ to 0.1 in our experiments. Our RFNet-4D was trained with a batch size of 16, and on a single NVIDIA RTX 3090 GPU. We evaluated all the variants of our network (see Ablation study) on the validation set for every 2,000 iterations during the training process, and used the best model of each variant on the validation set for evaluation of the variant on the test set. The training process was completed once there were no further improvements achieved. For calculating the losses during training, we randomly sampled a fixed number of 512 points in 3D space and time interval for reconstruction loss, and uniformly sampled trajectories of 100 points for flow estimation loss. More details can be found in our supplied code.

We also followed the evaluation setup used in [31]. Specifically, for each evaluation, we carried out two case studies: seen individuals but unseen motions (i.e., test subjects were included in the training data but their motions were not given in the training set), and unseen individuals but seen motions (i.e., test subjects were found only in the test data but their motions were seen in the training set).

Evaluation Metrics. To measure the performance of 4D reconstruction, we applied the common volumetric IoU (Intersection over Union) and the Chamfer distance reflecting the coincidence of reconstructed data and ground-truth data. To evaluate flow estimation, we used ℓ_2-distance to measure the correspondences between estimated flows and ground-truth flows.

4.2   Results

We report the performance of our RFNet-4D in two case studies in Table 1. As shown in experimental results, our method performed better in the seen individuals case study, for both object reconstruction and flow estimation. However, our method works consistently, and the differences in all performance metrics

Methods	Seen Individuals Unseen Motions			Unseen Individuals Seen Motions		
	IoU↑	Chamfer↓ $(\times 10^{-3})$	Corres.↓ $(\times 10^{-2})$	IoU↑	Chamfer↓ $(\times 10^{-3})$	Corres.↓ $(\times 10^{-2})$
PSGN-4D [10]	-	0.6189	1.1083	-	0.6877	1.3289
ONet-4D [27]	0.7712	0.5921	-	0.6827	0.7007	-
OFlow [31]	0.8172	0.1773	0.8699	0.7361	0.2741	1.0842
LPDC [38]	0.8511	0.1526	**0.7803**	0.7619	0.2188	0.9872
4DCR [15]	0.8171	0.1667	-	0.6973	0.2220	-
RFNet-4D	**0.8547**	**0.1504**	0.8831	**0.8157**	**0.1594**	**0.9155**

Table 1: **Quantitative evaluation** of our method and existing methods on seen and unseen individuals test splits, in both reconstruction and flow estimation task. We report the volumetric IoU (higher is better), Chamfer distance (lower is better) and correspondence ℓ_2 distance (lower is better). The notation '-' means no results, e.g., PSDN-4D does not perform reconstruction, ONet-4D and 4DCR do not predict point correspondences across time. For each evaluation metric, best performance is highlighted.

between the two case studies are marginal. For instance, the IoU difference between the two case studies is less than 4%, the differences in Chamfer distance and ℓ_2 correspondence between these case studies are about 0.009×10^{-3} and 0.03×10^{-2} respectively.

In addition to evaluation of our method, we also compared it with existing methods including PSGN-4D [10], ONet-4D [27], OFlow [31], LPDC [38], and 4DCR [15]. For the previous works, we used their published pre-trained models which had also been trained on the same training data from D-FAUST dataset. Note that we also re-trained the previous models using their released source code. However, we were not able to achieve the same results as reported in their papers. We show comparison results in Table 1. It can be seen that RFNet-4D outperforms all existing works in 4D reconstruction using both IoU and Chamfer distance metrics. For flow estimation, RFNet-4D achieves comparable performance with state-of-the-art on seen individuals, e.g., there is a slight difference (about 0.1×10^{-2}) in ℓ_2-distance from the first ranked method (i.e., LPDC). However, our RFNet-4D is trained using unsupervised fashion requiring no labels for learning point correspondences, while existing works follow supervised paradigm. Furthermore, as shown in Table 1, our method stands first in flow estimation on unseen individuals sequences, showing its robustness to novel object shapes.

We visualise several results of our methods and existing ones in Fig. 5. To illustrate these results, we apply the Multiresolution IsoSurface Extraction (MISE) algorithm [27] and Marching Cubes algorithm [25] on reconstructed occupancy maps to generate surface meshes. Compared with existing methods, our RFNet-

4D achieves higher reconstruction quality with better geometry recovery, e.g., the reconstructed hands produced our method are more complete. In addition, by coupling both spatial and temporal information, the poses of body parts, e.g., the head, the lower arms, are well reserved by our method (in reference to corresponding ground truth meshes). The results also show that our method is better than existing works (e.g., OFlow, LPDC) in flow estimation, as clearly shown in the predicted flows in the two hands. More qualitative results are provided in the supplementary material.

4.3   Ablation Studies

In this section, we present ablation studies to verify different aspects in the design of our model. In particular, we verified our joint learning of spatio-temporal representations for 4D point clouds reconstruction and flow estimation, compared with tackling these two tasks independently. We proved the improvement of learning flows in both forward and backward direction. We compared different distance metrics including the sliced Wasserstein distance (SWD) and the Hausdorff distance (HD), for the implementation of the flow loss in Eq. (1), and proved that our choice, i.e., the Chamfer distance, is the best. We experimented our model in both unsupervised and supervised fashion though it is intentionally designed for unsupervised learning, showing the flexibility our model.

We summarise results of our ablation studies in Table 2. Note that, in each ablation study, only one change was applied at a time while other settings remained unchanged. For settings using either temporal or spatial information (see the first two rows in Table 2), only the corresponding encoder and decoder (i.e., spatial/temporal encoder and decoder) were activated while the counterpart encoder and decoder were frozen. These settings correspond to solving the flow estimation and reconstruction task separately. To experiment our RFNet-4D with supervised learning for flow estimation, we followed the settings used in OFlow [31]. In particular, we used ground-truth point correspondences from the training data and ℓ_2 distance for the motion loss, i.e., replacing the Chamfer distance in Eq. (1) by ℓ_2 distance. Note that, D-FAUST dataset is fully annotated with point correspondences and thus also supports supervised learning. When training our model in unsupervised manner, those point correspondences were not used. Experimental results in Table 2 clearly confirm the design of our RFNet-4D in both object reconstruction and flow estimation.

4.4   Complexity Analysis

In this section, we provide a complexity analysis on the memory footprint and computational efficiency of our RFNet-4D and several existing models including OFlow [31] and LPDC [38] (current state-of-the-art). In this experiment, we trained all the models with a batch size of 16, using a sequence of 17 time steps with consistent intervals. All the models were run on a single NVIDIA RTX 3090. We report the memory footprint, training and inference time in Table 3. For the training time, we computed the average of batch training time throughout the

Variant	IoU $\uparrow$	Chamfer $(\times 10^{-3})\downarrow$	Corr. $(\times 10^{-2})\downarrow$
RFNet-4D (only temporal flows)	-	-	1.5519
RFNet-4D (only spatial points)	0.7712	0.5921	-
RFNet-4D (only FW motion)	0.4988	2.4887	3.5868
RFNet-4D (SWD loss)	0.4305	4.4621	4.0711
RFNet-4D (HD loss)	0.7953	0.2103	1.3017
RFNet-4D (supervised)	**0.8656**	**0.0927**	**0.8125**
RFNet-4D (unsupervised)	0.8547	0.1504	0.8831

Table 2: **Ablation studies** on of various settings of our method. For each evaluation metric, best performance is highlighted.

Method	Memory	Training (sec/iter)	Inference (sec/seq)
OFlow [31]	3.96GB	4.65s	0.95s
LPDC [38]	11.90GB	2.09s	0.44s
RFNet-4D	14.20GB	1.33s	0.24s

Table 3: **Space and time complexity** of our method and existing ones.

first 100k iterations of training (seconds per iteration). For the inference time, we reported the average time required to infer using a batch size of 1 for 1k test sequences (seconds per sequence). As shown in Table 3, despite our model takes larger memory footprint for training, its training time is approximately 3.5 times and 1.6 times faster than that of OFlow and LPDC respectively. Similarly, our model performs 1.9 times and 4 times faster than OFlow and LPDC in inference. We found that OFlow take much longer time for training since OFlow makes use of an ODE-solver requiring intensive computations and gradually increasing the number of iterations to fulfill error tolerance.

5   Discussion and Conclusion

This paper proposes RFNet-4D, a network architecture for jointly reconstruction of objects and estimation of temporal flows from dynamic point clouds. The proposed network is built upon a compositional encoder effectively capturing informative spatio-temporal representations for 4D point clouds, and a joint learning paradigm leveraging sub-tasks to improve overall performance. We extensively evaluated our proposed RFNet-4D and compared it with existing works on benchmark dataset. Experimental results demonstrated the effectiveness and efficiency of our method in comparison with current state-of-the-art.
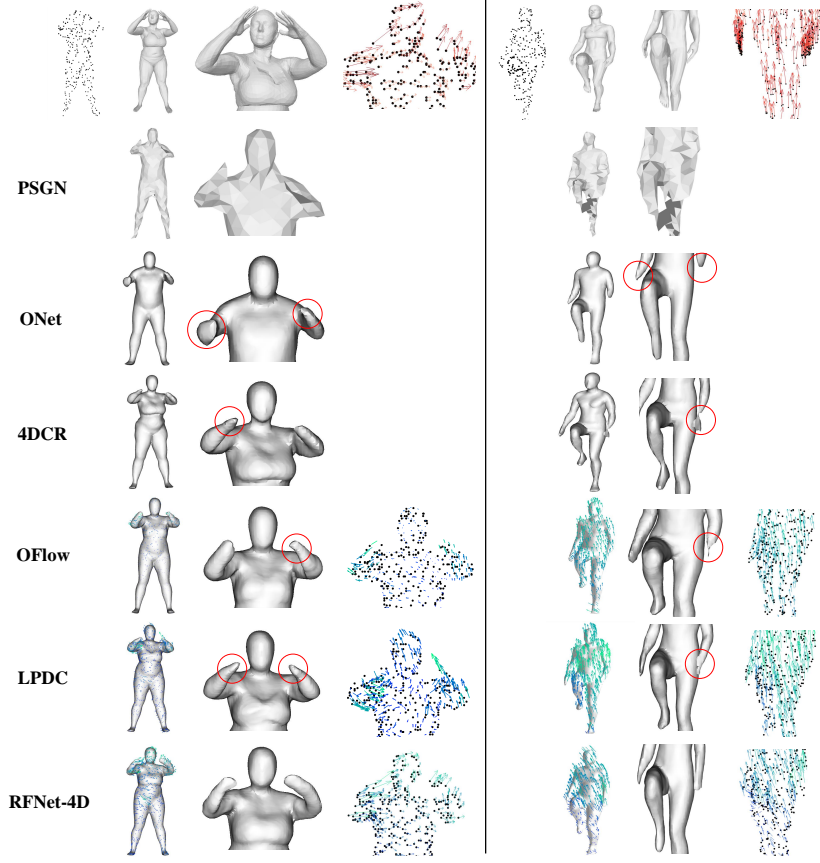
Fig. 5: **Qualitative evaluation** of our method and existing methods. The first row includes (from left to right): input point cloud, ground truth mesh of entire body, ground truth mesh of upper/lower body, and ground-truth flows (darker vectors show stronger motions). Each following row represents corresponding reconstruction and flow estimation results. Severe errors are highlighted.

There is also room for future research. First, we found that existing 4D reconstruction methods often suffer from large displacements between data frames. Second, their reconstruction quality tends to drop over time due to accumulated errors. It is also worthwhile to study 4D reconstruction for different types of objects, and with more challenging input data types, e.g., LiDAR point clouds that are commonly used in autonomous driving applications.

References

1. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video based reconstruction of 3D people models. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2

2. Biasotti, S., Cerri, A., Bronstein, A., Bronstein, M.: Recent trends, applications, and perspectives in 3D shape similarity assessment. Comput. Graph. Forum (2016) 5

3. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 10

4. Chabra, R., Lenssen, J.E., Ilg, E., Schmidt, T., Straub, J., Lovegrove, S., Newcombe, R.: Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In: European Conference on Computer Vision (ECCV) (2020) 4

5. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1

6. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3D shape reconstruction and completion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4

7. Coskun, H., Achilles, F., DiPietro, R., Navab, N., Tombari, F.: Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2017) 2

8. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. In: International Conference on Learning Representations (ICLR) (2017) 8

9. Erler, P., Guerrero, P., Ohrhallinger, S., Mitra, N.J., Wimmer, M.: Points2Surf: Learning implicit surfaces from point clouds. In: European Conference on Computer Vision (ECCV) (2020) 4

10. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3D object reconstruction from a single image. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4, 8, 11

11. Girdhar, R., Fouhey, D., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: the European Conference on Computer Vision (ECCV) (2016) 4

12. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 15 (2011) 8

13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 7, 8

14. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. IEEE/CVF International Conference on Computer Vision (ICCV) (2017) 5

15. Jiang, B., Zhang, Y., Wei, X., Xue, X., Fu, Y.: Learning compositional representation for 4D captures with neural ODE. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4, 8, 11

16. Jiang, C.M., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T.: Local implicit grid representations for 3D scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4

17. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: European Conference on Computer Vision (ECCV) (2018) 4
18. Kanazawa, A., Zhang, J.Y., Felsen, P., Malik, J.: Learning 3D human dynamics from video. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2
19. Kato, H., Ushiku, Y., Harada, T.: Neural 3D mesh renderer. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
20. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson Surface Reconstruction. In: Symposium on Geometry Processing (2006) 4
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015) 10
22. Leroy, V., Franco, J.S., Boyer, E.: Multi-view dynamic shape refinement using local temporal integration. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2017) 2, 4
23. Liao, Y., Donné, S., Geiger, A.: Deep marching cubes: Learning explicit surface representations. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4
24. Liu, X., Yan, M., Bohg, J.: MeteorNet: Deep learning on dynamic 3D point cloud sequences. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 7
25. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. SIGGRAPH Comput. Graph. (1987) 4, 11
26. Lüthi, M., Gerig, T., Jud, C., Vetter, T.: Gaussian process morphable models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018) 5
27. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3D reconstruction in function space. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 4, 8, 11
28. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.: Implicit surface representations as layers in neural networks. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 1
29. Mustafa, A., Kim, H., Guillemaut, J.Y., Hilton, A.: General dynamic scene reconstruction from multiple view video. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2015) 2, 4
30. Mustafa, A., Kim, H., Guillemaut, J.Y., Hilton, A.: Temporally coherent 4d reconstruction of complex dynamic scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 2, 4
31. Niemeyer, M., Mescheder, L.M., Oechsle, M., Geiger, A.: Occupancy flow: 4D reconstruction by learning particle dynamics. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 2, 4, 5, 7, 8, 10, 11, 12, 13
32. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 4
33. Pekelny, Y., Gotsman, C.: Articulated object reconstruction and markerless motion capture from depth video. Computer Graphics Forum (2008) 2
34. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: European Conference on Computer Vision (ECCV) (2020) 4
35. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3D object detection from RGB-D data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 4

36. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3D classification and segmentation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 7

37. Slavcheva, M., Baust, M., Ilic, S.: Towards implicit correspondence in signed distance field evolution. In: IEEE/CVF International Conference on Computer Vision (ICCV) Workshops (2017) 5

38. Tang, J., Xu, D., Jia, K., Zhang, L.: Learning parallel dense correspondence from spatio-temporal descriptors for efficient and robust 4D reconstruction. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 5, 7, 8, 11, 12, 13

39. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. IEEE/CVF International Conference on Computer Vision (ICCV) (2017) 4

40. Tung, H.Y., Tung, H.W., Yumer, E., Fragkiadaki, K.: Self-supervised learning of motion capture. In: Advances in Neural Information Processing Systems (2017) 2

41. de Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., Courville, A.C.: Modulating early visual processing by language. In: Advances in Neural Information Processing Systems (2017) 8

42. Wand, M., Jenke, P., Huang, Q., Bokeloh, M., Guibas, L., Schilling, A.: Reconstruction of deforming geometry from time-varying point clouds. In: Proceedings of the Fifth Eurographics Symposium on Geometry Processing (2007) 2

43. Wang, J., Wen, C., Fu, Y., Lin, H., Zou, T., Xue, X., Zhang, Y.: Neural pose transfer by spatially adaptive instance normalization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 5

44. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2Mesh: Generating 3D mesh models from single rgb images. In: European Conference on Computer Vision (ECCV) (2018) 4

45. Wang, P.S., Liu, Y., Guo, Y.X., Sun, C.Y., Tong, X.: O-CNN: Octree-based convolutional neural networks for 3d shape analysis. ACM Trans. Graph. (2017) 4

46. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2Mesh++: Multi-view 3D mesh generation via deformation. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 4

47. Zheng, Q., Fan, X., Gong, M., Sharf, A., Deussen, O., Huang, H.: 4D reconstruction of blooming flowers. Computer Graphics Forum (2017) 2