GarmentTracking: Category-Level Garment Pose Tracking

Han Xue[1,2], Wenqiang Xu[2], Jieyi Zhang[2], Tutian Tang[2], Yutong Li[2], Wenxin Du[2], Ruolin Ye[3], Cewu Lu[†1,2]
[1]Shanghai Qi Zhi Institute [2]Shanghai Jiao Tong University [3] Cornell University
[2]{xiaoxiaoxh, vinjohn, yi_eagle, tttang, davidliyutong, mnkmYuki, lucewu}@sjtu.edu.cn
[3]ry273@cornell.edu

Abstract

*Garments are important to humans. A visual system that can estimate and track the complete garment pose can be useful for many downstream tasks and real-world applications. In this work, we present a complete package to address the category-level garment pose tracking task: (1) A recording system **VR-Garment**, with which users can manipulate virtual garment models in simulation through a VR interface. (2) A large-scale dataset **VR-Folding**, with complex garment pose configurations in manipulation like flattening and folding. (3) An end-to-end online tracking framework **GarmentTracking**, which predicts complete garment pose both in canonical space and task space given a point cloud sequence. Extensive experiments demonstrate that the proposed GarmentTracking achieves great performance even when the garment has large non-rigid deformation. It outperforms the baseline approach on both speed and accuracy. We hope our proposed solution can serve as a platform for future research. Codes and datasets are available in https://garment-tracking.robotflow.ai.*

1. Introduction

Garments are one of the most important deformable objects in daily life. A vision system for garment pose estimation and tracking can benefit downstream tasks like MR/AR and robotic manipulation [7, 17]. The category-level garment pose estimation task is firstly introduced in GarmentNets [11], which aims to recover the full configuration of an **unseen** garment from a single static frame. Unlike the non-rigid tracking methods [9, 10, 16, 19, 27, 34, 35] which can only recover the geometry of the visible regions, pose estimation task can also reconstruct the occluded parts of the object. Another line of works [14, 15, 21, 22, 28–30] (non-rigid 4D reconstruction which can reconstruct complete object geometry) cannot be directly applied on garments, since they assume the object has a watertight geometry. In contrast, garments have thin structures with holes.

In this paper, we propose a new task called **Category-level Garment Pose Tracking**, which extends the single-frame pose estimation setting in [11] to pose tracking in dynamic videos. Specifically, we focus on the pose tracking problem in garment manipulation (*e.g.* flattening, folding). In this setting, we do not have the priors of the human body like previous works for clothed humans [18,26,31,41]. Therefore, we must address the extreme deformation that manipulated garments could undergo.

To tackle the garment pose tracking problem, we need a dataset of garment manipulation with complete pose annotations. However, such a dataset does not exist so far to the best of our knowledge. To build such a dataset, we turn to a VR-based solution due to the tremendous difficulty of garment pose annotation in the real world [10]. We first create a real-time VR-based recording system named **VR-Garment**. Then the volunteer can manipulate the garment in a simulator through the VR interface. With VR-Garment, we build a large-scale garment manipulation dataset called **VR-Folding**. Compared to the single static garment configuration (*i.e.* grasped by one point) in GarmentNets, our manipulation tasks include *flattening* and *folding*, which contain much more complex garment configurations. In total, our VR-Folding dataset contains 9767 manipulation videos which consist of 790K multi-view RGB-D frames with full garment pose and hand pose annotations on four garment categories selected from the CLOTH3D [8] dataset.

With the VR-Folding dataset, we propose an end-to-end online tracking method called **GarmentTracking** to perform category-level garment pose tracking during manipulation. For the garment pose modeling, we follow GarmentNets [11] to adopt the normalized object coordinate space (NOCS) for each category. Nevertheless, tracking garment pose raises new challenges compared to single-frame pose estimation: (1) How to fuse inter-frame geometry and correspondence information? (2) How to make the tracking prediction robust to pose estimation errors? (3) How to achieve

† Cewu Lu is the corresponding author, the member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China and Shanghai Qi Zhi Institute.

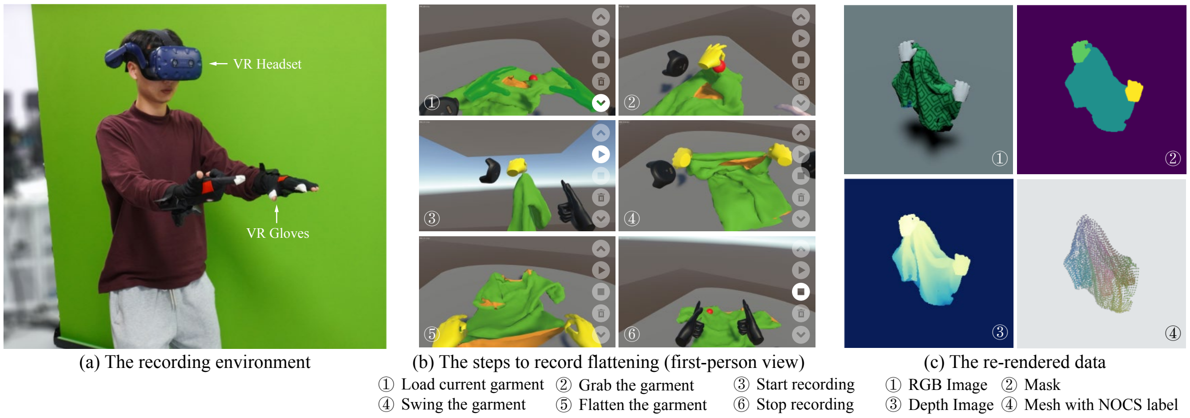

Figure 1. **The pipeline of our Virtual Realty recording system (VR-Garment).** (a) A volunteer needs to put on a VR headset and VR gloves. (b) By following the guidance of a specially designed UI, the volunteer begins to collect data efficiently. (c) After recording, we re-render multi-view RGB-D images with Unity [6] and obtain masks and deformed garment meshes with NOCS labels.

tracking in real-time? To address these challenges, we conduct GarmentTracking in three stages, namely *NOCS predictor*, *NOCS refiner*, and *warp field mapper*. Firstly, it predicts per-frame features and fuses them for canonical coordinate prediction. Then it refines the predicted canonical coordinates and the geometry with a NOCS refiner to reduce the accumulated errors. Finally, it maps the prediction in canonical space to the task space (*i.e*. coordinate frame of the input point cloud).

Since no previous work is designed for the tracking setting, we use GarmentNets [11] as a single-frame prediction baseline for comparison. We also perform extensive ablative experiments to reveal the efficacy of our design choices. Finally, we collect real-world data on garment manipulation and show the qualitative results of our method. In our design, we avoid the computationally expensive Marching Cubes [25] for reconstructing the canonical mesh frame by frame, so that we can achieve tracking at 15 FPS with an RTX 3090 GPU (**5 times faster** than the baseline approach).

We summarize our contributions as follows:

1). We propose a VR-based garment manipulation recording system named **VR-Garment**. It can synchronize human operations into the simulator and collect garment manipulation data.

2). We propose a large-scale garment manipulation dataset named **VR-Folding** for pose tracking. During manipulation, garments exhibit diverse configurations.

3). We propose a real-time end-to-end framework named **GarmentTracking** for category-level garment pose tracking. It can serve as a strong baseline for further research. We also demonstrate its generalization ability to real-world garment recordings with models trained by simulated data.

2. Related Work

Category-level Object Pose Estimation and Tracking. Object pose is the configuration of the object posited in the observation space. For the rigid object, we can describe its pose in 6 degrees of freedom (DOFs), *i.e*. 3 for translation and 3 for rotation. However, for the non-rigid object, like garments, the object pose can be of near-infinite DOFs.

On the other hand, the category-level object pose estimation task aims to learn a model that can predict unseen object poses of the same category [20, 23, 24, 38, 39]. The concept is first introduced to estimate rigid object pose [38]. In [38], Wang *et al*. proposed a Normalized Object Coordinate Space (NOCS) as a category-specific canonical representation. Following the idea, Li *et al*. [20] proposed a hierarchical NOCS representation for articulated objects.

To handle the near-infinite DOF nature and the category-level generalization requirement, GarmentNets [11] also defined NOCS for each garment category. They predicted the garment pose by mapping the reconstructed mesh from canonical space to task space. However, GarmentNets [11] treats each frame individually, which hampers its stability for inter-frame prediction, and its ability to infer complex poses from sequential movements. Our GarmentTracking is proposed for these tracking issues.

Non-rigid Tracking and Reconstruction. Tracking and reconstructing non-rigid deforming objects is an important research area in computer vision and graphics. One line of works [9, 10, 16, 19, 27, 34, 35] perform free-form tracking, which does not assume any geometric prior. For example, DynamicFusion [27] used a hierarchical node graph structure and an efficient GPU solver to reconstruct the visible surface of the object. Deepdeform [10] and Bozic *et al*. [9] leveraged learning-based correspondences to track deformable objects. However, unlike pose estimation methods which reconstruct the complete configuration of objects, all these methods can not reconstruct occluded parts.

Another line of works [14, 15, 21, 22, 28–30] can perform 4D reconstruction from RGB-D videos, which captures the complete geometry of the object both in space and time. Unfortunately, the shape representations of these methods have limitations when adapting to garment pose reconstruction under large deformations. For example, Fusion4D [15], Motion2fusion [14], 4DComplete [21], OcclusionFusion [22], NPMs [29] and OccupancyFlow [28] heavily rely on watertight object modeling such as SDF, TSDF, or occupancy grids to reconstruct object surfaces. Such modeling is not suitable for objects with open and thin structures like garments.

Garment-related Dataset. Current garment-related datasets can be divided into *asset datasets* [8, 41] and *task datasets* [10, 11, 26, 37]. *Asset datasets* provide garment models for different tasks. For example, GarmentNets [11] proposed a simulation dataset for category-level pose-estimation task based on CLOTH3D [8]. We also build our VR-Folding dataset based on [8]. Other task datasets do not require complete garment models [26]. For example, CAPE [26] deals with the clothed human reconstruction task. However, the human body limits the possible garment states. DeepDeform [10] dataset contains simple scenes where a person lifts one garment with minor deformations, and it only annotates sparse keypoint correspondences between frames. A real-world cloth-folding dataset proposed by Verleysen *et al*. [37] contains videos of cloth-folding actions, but it only annotates the contour of garments in 2D images. Our VR-Folding dataset is the first dataset designed for category-level garment pose tracking in manipulation, and it contains dynamic scenes which include complex human actions and garment configurations.

3. VR-Folding Dataset

To build the VR-Folding dataset, we develop a real-time data recording system called **VR-Garment** (Fig. 1). In this way, we can combine human experience and the benefits of simulation environments (*i.e*. easy access to ground-truth poses) to efficiently collect a large amount of data with natural and complex poses. We will describe the system design and the operation procedures in Sec. 3.1 and Sec. 3.2. Then we will describe the data statistics in Sec. 3.3.

3.1. VR-Garment

In this section, we will first describe the hardware and software setup of the system and then the recording procedure for volunteers to operate. The recording system is illustrated in Fig. 1.

Hardware. On the hardware side, our recording system needs an HTC Vive Pro [1] VR headset and Noitom Hi5 [2] VR gloves which can track finger poses in the real world and reproduce them with virtual hands in the simulator through a VR interface.

Software. We developed our VR recording framework based on Unity [6] for its good support of mainstream VR devices. The cloth physics simulation in Unity is achieved by Obi [3]. Specifically, we implemented a simple UI and a grasping system in VR, which allow users to grasp or release any point on the garment surface when two virtual fingertips make contact with the garment.

Recording Procedure. Firstly, a volunteer must wear a VR headset and VR gloves. Then he should observe a garment instance randomly dropping on a table in Unity. Next, he performs a pre-defined manipulation task (*e.g*. folding). In the manipulation process, we save the deformed garment mesh and hand poses for each frame. When the task is done, the volunteer will use special gestures (*e.g*. fist) to send commands for moving to the next garment instance. After recording, we re-render multi-view RGB-D images in Unity with the saved animation data and generate corresponding ground-truth annotations (*e.g*. garment poses, masks). Note that all the rendering settings (*e.g*. lights, cameras, textures *etc*.) can be customized even after recording.

3.2. Task Definition

In a typical cloth folding process, we operate this task in two stages, namely *flattening* and *folding*:

Flattening: Firstly, a garment will drop on the virtual table in Unity. Then our system will randomly choose one point on the garment surface. Next, the volunteer will grasp that point with one hand and lift the garment in the air (an initial configuration similar to that in GarmentNets [11]). Next, the volunteer will try to grasp and fling the garment repeatedly with two hands until the garment is smoothed and in a flattened T-pose (see Fig. 1). Please see the supplementary files for more details about the task.

Folding: Firstly, a garment in a flattened T-pose will be placed on the virtual table. Then the volunteer will repeatedly perform pick-and-place actions with both hands until the garment is folded. Though people may have different preferred steps to achieve folding, we have defined general rules for each category and asked the volunteers to follow them. Please see the supplementary files for more details.

3.3. Data Statistics

All the garment meshes used in our system are from CLOTH3D [8]. We choose 4 categories from CLOTH3D dataset, namely *Shirt*, *Pants*, *Top* and *Skirt*. For *flattening* task, we recorded 5871 videos which contain 585K frames in total. For *folding* task, we recorded 3896 videos which contain 204K frames in total. As shown in Fig. 1, the data for each frame include multi-view RGB-D images, object masks, full garment meshes, and hand poses. Please see the supplementary files for more statistics on the dataset.

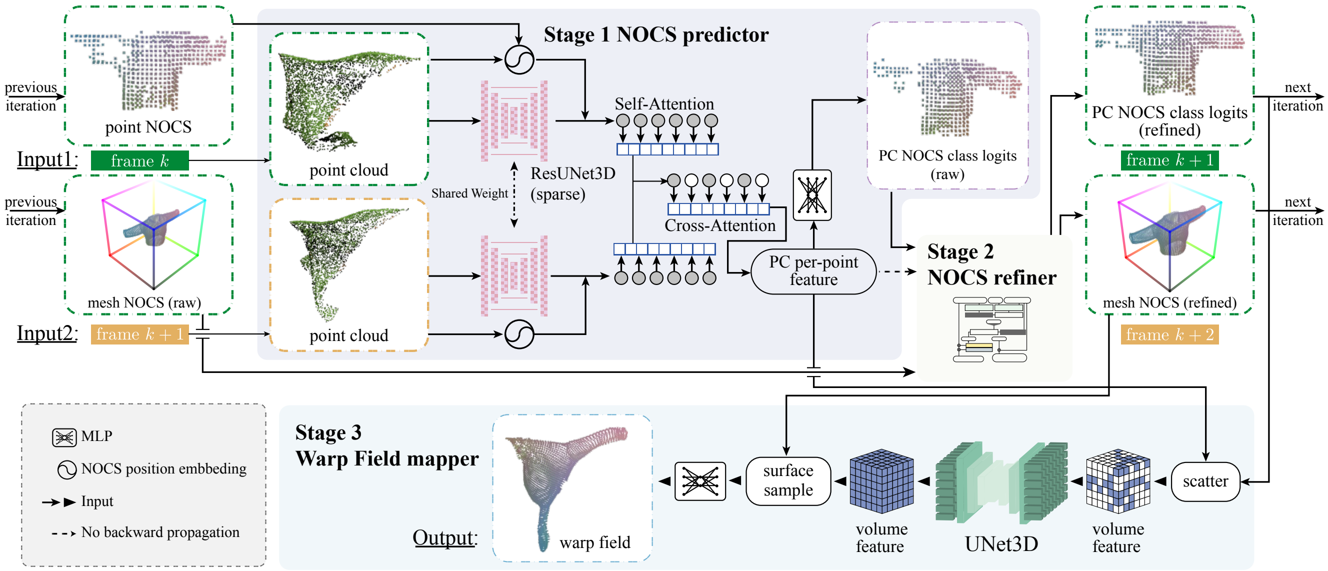

Figure 2. The overview of **GarmentTracking**. Given the per-point NOCS coordinate of the first frame and a rough canonical shape (mesh NOCS), our tracking method takes two frames of the partial point cloud as input. In stage 1, the NOCS predictor will generate an inter-frame fusion feature and predict raw NOCS coordinates. In stage 2, the NOCS refiner will refine the NOCS coordinates and the canonical shape simultaneously. In stage 3, the warp field mapper will predict the warp field which maps from canonical space to task space.

4. Method

This paper proposes an end-to-end online tracking method called **GarmentTracking** for category-level garment pose tracking. As shown in Fig. 2, given a first-frame garment pose (point NOCS, *i.e*. canonical coordinates of partial point cloud) and a rough canonical shape (mesh NOCS, *i.e*. sampled points from mesh in canonical space) of an instance, it takes point cloud sequences as input and generates complete garment geometry with inter-frame correspondence (*i.e*. NOCS coordinates). Specifically, GarmentTracking can be divided into three stages. In the first stage (Sec. 4.1), the network will predict canonical coordinates for the partial input point cloud. In the second stage (Sec. 4.2), the network will refine the predicted canonical coordinates and the input canonical shape. In the third stage (Sec. 4.3), the network will use the refined canonical shape and canonical coordinates to predict a warp field that maps from canonical space to task space (*i.e*. the coordinate frame of the input point cloud).

4.1. Canonical Coordinate Prediction

4.1.1 Normalized Garment Canonical Space

Following the definition of garment representation in GarmentNets [11], we use Normalized Object Coordinate Space (NOCS) coordinates as an intermediate representation for object states in a category. As shown in Fig. 2, the rest state of a garment is the T-pose defined by the garment worn by a person (provided by CLOTH3D [8] dataset).

4.1.2 3D Feature Extractor

The thin structure and near-infinite DOF nature of garments may result in many complicated poses (*e.g*. multi-layer cloth stacked together) that require feature extraction for granular local details. Our method uses a high-resolution sparse 3D convolution network (ResUNet3D) proposed by FCGF [12] to extract the per-point feature from the raw point cloud. ResUNet3D is a UNet-like network with skip connections and residual blocks. Please refer to the supplementary files for further details of the network.

4.1.3 Inter-frame Feature Fusion with Transformer

After extracting the feature from the extractor, we apply the inter-frame feature fusion with Transformer [36].

Feature Matching Inspired by PTTR [40], we perform feature matching with self-attention and cross-attention modules based on Transformer [36]. In general, we first use a self-attention module to individually aggregate point features for the two input frames. Then we use a cross-attention module to perform feature matching between two frames. Intuitively, the self-attention operation can have a global understanding of the current frame, and the cross-attention operation can capture cross-frame correlations and generate a relation-enhanced fusion feature. The self-attention and cross-attention modules are based on the relation attention module (RAM) proposed by PTTR [40]. Please see the supplementary files for more details on the relation attention module.

NOCS Prediction After obtaining the per-point fusion feature via the cross-attention module, we predict the per-point canonical coordinate with MLP. We follow GarmentNets [11] and formulate this problem as a classification task instead of a regression task. Specifically, we divide each axis into 64 bins and the network independently predicts each axis's classification score. During training, we use a cross-entropy loss to supervise the classification scores.

NOCS Coordinates for Positional Embedding We have per-point NOCS coordinate prediction from the previous frame, which contains clearer geometric and structural information. We use it for positional embedding [36], which will be added to input features before feeding into transformers. The positional embeddings for two frames are calculated as Eq. (1):

$$\mathbf{emb}_1 = f_1([\mathbf{P}_1^{xyz}, \mathbf{P}_1^{nocs}]), \mathbf{emb}_2 = f_2([\mathbf{P}_2^{xyz}]), \quad (1)$$

where $\mathbf{P}_1^{xyz}$ and $\mathbf{P}_2^{xyz}$ are the partial input point clouds of the two frames, and $\mathbf{P}_1^{nocs}$ is the predicted per-point NOCS coordinates of the partial point cloud in the previous frame. Here $f_1(\cdot)$ and $f_2(\cdot)$ are learned MLP. By fusing NOCS coordinates into positional embedding, the transformer network will incorporate positional and semantic information from previous frames. Besides, empirically speaking, utilizing intermediate representations like NOCS coordinates instead of complete garment poses can increase the robustness against noisy predictions during long-term tracking.

4.2. NOCS Refiner

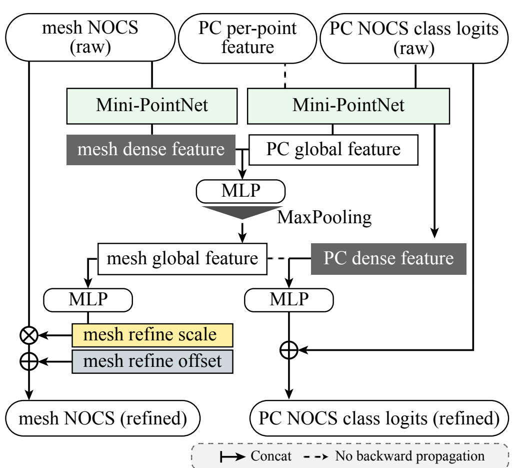

Figure 3. PC-Mesh Fusion Refiner

Since the canonical shape can be generated by other methods like GarmentNets [11], or augmented with noise, it might be inaccurate. On the other hand, the NOCS coordinate predictions can also be noisy. Such inaccuracy could cause errors to be accumulated during tracking. To mitigate this problem, we propose a NOCS PC (Point Cloud)-Mesh intertwined refiner, **NOCS refiner**. As shown in Fig. 3, the predicted NOCS coordinates can reveal the cues of the input point cloud, such as scales and offsets, while the canonical shape can provide information about the complete geometry. Thus they can complement each other. We describe the NOCS refiner in two parts (*PC refiner* and *Mesh refiner*):

PC Refiner: Firstly, the predicted NOCS classification scores and the per-point fusion feature from the transformer will be concatenated and fed into a Mini-Pointnet [32]. Next, the dense feature will be fused with the global mesh feature generated by *Mesh Refiner* with concatenation. Finally, we use MLP to predict the final delta logits with the fused dense feature. We use cross-entropy loss to supervise the refined classification logits during training.

Mesh Refiner: Firstly, we use a Mini-Pointnet to extract dense features from raw canonical shapes. Then we concatenate the dense mesh feature generated by *PC Refiner* with the global feature from the partial point cloud to obtain the fused dense feature. Next, we use an MLP with global pooling to extract the final global shape feature. Finally, we predict the global scale factor and offset for the canonical shape with the global shape feature and an MLP. Finally, we use L2 loss to supervise the refined mesh points during training.

4.3. Warping from Canonical To Task Space

4.3.1 Feature Scattering with Canonical Coordinates

After obtaining the refined canonical (NOCS) coordinate prediction (Sec. 4.2) of a partial point cloud, we scatter the per-point feature generated by Transformer (Sec. 4.1) into a 32^3 feature volume. The "scatter" operation is performed by copying the feature vector to the target location in volume with predicted NOCS coordinates. All features mapped to the same volume index will be aggregated with a channel-wise maximum operation. And all the volume locations with no corresponding feature vectors are filled with zeros. Then the feature volume will be fed into a 3D UNet [13], then we can obtain a dense feature volume $\mathcal{V}$ for warp field prediction.

4.3.2 Warp Field Prediction

Finally, we map the refined canonical shape (Sec. 4.2) from canonical space to task space. The output contains the full configuration of the garment, including the occluded parts. It is achieved by warp field prediction [11], which is an implicit neural function $w(p; \mathcal{V}) \in \mathbf{R}^3$ that takes a query point p in the canonical space as input and infers the corresponding location of p in task space. Here $w(\cdot)$ is a learned MLP. We use L2 loss to supervise the warp field prediction. In training, the query points are sampled from the canonical mesh surface. In inference, the query points are generated by our Mesh Refiner (Sec. 4.2).

5. Experiments

5.1. Implementation Details

We implement our method with Pytorch [4] and use Adam optimizer with a learning rate of 0.0001. The training stage takes about 150 epochs to converge, which lasts for 1-3 days on an RTX 3090 GPU, depending on the training dataset sizes for different categories. We randomly sample 4000 points from the input partial point cloud and 6000 points from the input canonical mesh surface for each frame. During training, we randomly add noise to the partial point-cloud canonical coordinates $\mathbf{P}_1^{nocs}$ of the previous frame by randomly generating a NOCS scale factor $\mathbf{s}_{pc} \in [0.8, 1.2]^3$ and a global NOCS offset $\mathbf{o}_{pc} \in [0, 0.1]^3$. We also add noise to the input canonical mesh by randomly generating a global NOCS scale factor $\mathbf{s}_{mesh} \in [0.8, 1.2]^3$ during training. Please see the supplementary files for further details on training, inference, and network structure.

5.2. Metrics

NOCS Coordinate Distance (D_{nocs}). We calculate the point-wise L2 distance between the predicted NOCS coordinate of the partial point cloud with the ground-truth NOCS labels. This metric evaluates the quality of per-point NOCS coordinate prediction for input partial point cloud.

Chamfer Distance (D_{chamf}). We calculate the Chamfer distance in centimeters between the reconstructed mesh points and the ground-truth mesh points in task space. This metric can evaluate the quality of *surface reconstruction*.

Correspondence Distance (D_{corr}, A_d). We calculate point-wise L2 distance in centimeters between the reconstructed mesh and the ground-truth mesh for each frame in task space. The correspondences are based on the NOCS coordinates (*i.e*. each point on the predicted mesh will find the closest point on the ground-truth mesh in NOCS). This metric can evaluate the quality of garment *pose estimation*. In practice, we find the variance of the error distribution in different frames is very large, which makes the mean correspondence distance D_{corr} across all frames dominated by the worst cases. So we additionally introduce A_d which represents the accuracy (*i.e*. ratio of frames) with $D_{corr} < d$.

5.3. Experiment Results

5.3.1 Main Experiments

Baselines In Tab. 1, we compare GarmentNets with two settings of our method:

GarmentNets [11]: As the only-existing method for category-level garment pose estimation, GarmentNets focused on the single-frame setting. We adapt it for tracking by prediction frame by frame.

Ours (GT): Our tracking method given the ground-truth first-frame garment pose and the ground-truth canonical mesh as initialization.

Ours (Pert.): Our tracking method when the first-frame garment pose and the input canonical shape are perturbed with noise. Specifically, we use the same noise distribution in training (Sec. 5.1) which adds global NOCS scale and offset to the first-frame canonical coordinates of partial point-cloud and the canonical mesh. Additionally, we add per-point Gaussian noise (δ=0.05) to the input canonical coordinates of the first frame during inference.

Results Tab. 1 summarizes the quantitative results on the VR-Folding dataset. In general, our method outperforms GarmentNets in all metrics by a large margin. On the challenging A_{3cm} metric in *Folding* task and A_{5cm} in *Flattening* task, GarmentNets has very low performance (*e.g*. 0.8% in *Shirt Folding*), while our method achieves much higher scores (*e.g*. 29.0% in *Shirt Folding*), which proves that our method can generate more accurate predictions in videos compared to GarmentNets. Our method also outperforms GarmentNets on mean correspondence distance D_{corr} and chamfer distance D_{chamf}, which proves that our method can do well in both *pose estimation* and *surface reconstruction* tasks. Even with perturbation on first-frame poses (Ours with Pert. in Tab. 1), our method only shows minor performance loss (*e.g*. 37.9% $\rightarrow$ 36.6% in *Top Folding*) compared to using ground-truth as first-frame pose.

We also present some qualitative results in Fig. 4 and Fig. 5. We can see from Fig. 5 that the prediction results of GarmentNets are very unstable because it performs mesh reconstruction for each frame individually and can not utilize the information from previous frames. Conversely, our method can leverage input canonical mesh and inter-frame information to predict more stable and accurate results. Besides, GarmentNets suffers from ambiguity brought by symmetry (*e.g*. take a front side as a back side), which hampers its ability to predict accurate canonical coordinates (see Fig. 4). In contrast, our method can predict much more accurate canonical coordinates (*e.g*. D_{nocs} 0.162 v.s. 0.039 for *Pants Folding* in Tab. 1).

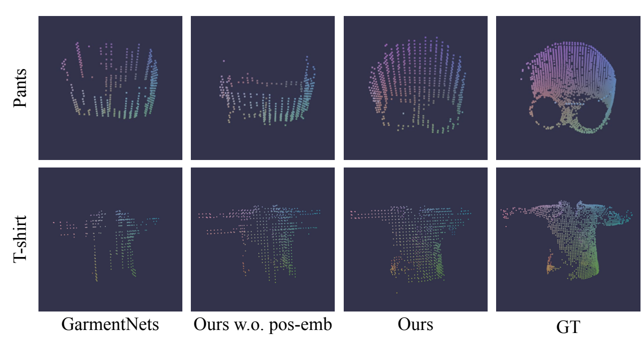

Figure 4. The canonical coordinate prediction results on the VR-Folding dataset.

Type	Method	Init.	Folding					Flattening				
			$A_{3cm}\uparrow$	$A_{5cm}\uparrow$	$D_{corr}\downarrow$	$D_{chamf}\downarrow$	$D_{nocs}\downarrow$	$A_{5cm}\uparrow$	$A_{10cm}\uparrow$	$D_{corr}\downarrow$	$D_{chamf}\downarrow$	$D_{nocs}\downarrow$
Shirt	GarmentNets [11]	N/A	0.8%	21.5%	6.40	1.58	0.221	13.2%	59.4%	10.54	3.54	0.135
	Ours	GT	**29.8%**	85.8%	**3.88**	**1.16**	**0.051**	**30.7%**	**83.4%**	**8.63**	**1.78**	**0.105**
	Ours	Pert.	29.0%	**85.9%**	3.88	1.18	0.052	25.4%	81.6%	8.94	1.85	0.109
Pants	GarmentNets [11]	N/A	16.2%	69.5%	4.43	1.30	0.162	1.5%	42.4%	12.54	4.19	0.185
	Ours	GT	**47.3%**	**94.0%**	**3.26**	**1.07**	**0.039**	**31.3%**	**78.2%**	**8.97**	**1.64**	**0.113**
	Ours	Pert.	42.8%	93.6%	3.35	1.10	0.039	30.7%	76.9%	9.55	2.71	0.143
Top	GarmentNets [11]	N/A	10.3%	53.8%	5.19	1.51	0.148	21.6%	57.6%	9.98	2.13	0.174
	Ours	GT	**37.9%**	**85.9%**	**3.75**	**0.99**	**0.051**	**36.5%**	**69.0%**	**9.41**	**1.59**	**0.113**
	Ours	Pert.	36.6%	86.1%	3.76	1.00	0.051	33.5%	68.1%	9.61	1.62	0.116
Skirt	GarmentNets [11]	N/A	1.1%	30.3%	6.95	1.89	0.239	0.1%	7.9%	18.48	5.99	0.287
	Ours	GT	**23.5%**	**71.3%**	**4.61**	**1.33**	**0.060**	**5.4%**	**39.4%**	**16.09**	**2.02**	**0.199**
	Ours	Pert.	22.8%	70.6%	4.72	1.36	0.060	2.3%	35.5%	16.55	2.15	0.207

Table 1. Quantitative results on VR-Folding dataset.

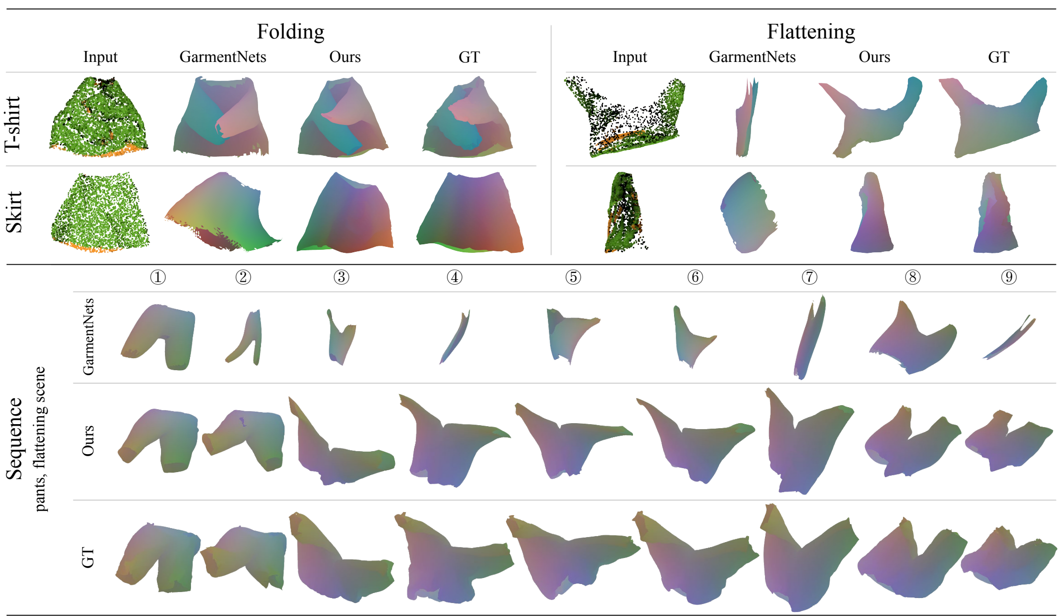

Figure 5. The qualitative results of pose estimation for **unseen** instances in VR-Folding dataset. In the long sequence tracking (shown in the lower part), our prediction still keeps high consistency with GT, while GarmentNets outputs a series of meshes that lack stability.

5.3.2 Ablation Study

NOCS Positional Embedding. In our method, NOCS positional embedding (Sec. 4.1.3) is the crucial design choice to leverage inter-frame correspondence information. As shown in Fig. 4, if we remove the NOCS positional embedding from our network, the network will suffer from the same ambiguity problem as GarmentNets due to symmetry.

NOCS Refiner. Unlike rigid object tracking, garment tracking has a higher demand for avoiding error accumulation in long videos, because the error distribution of predicted pose in testing can be very different from that in training due to the near-infinite DOF. As shown in Tab. 2, our proposed PC Refiner (Sec. 4.2) greatly influences the performance due to its ability to refine NOCS coordinate predictions in each frame. Besides, the Mesh Refiner (Sec. 4.2) also contributes to a slight performance improvement, indicating that our network has more tolerance for canonical mesh errors than NOCS coordinate errors.

Feature Extractor. As shown in Tab. 2, after we replace our feature extractor (*i.e.* ResUNet3D [12] based on sparse 3D convolution) with PointNet++ [33], the overall performance drops a lot. Thus high-resolution 3D convolution network should be a better choice for this task.

5.3.3 Robustness

Robustness against Noise We test our method under different levels of noise perturbation by increasing the initial pose noise level described in Sec. 5.3.1 by 1 or 2 times. Specifically, we augment the point-cloud NOCS coordinates of the first frame with a global scaling factor s_{pc}, a

Type	Method	Folding					Flattening				
		$A_{3cm}\uparrow$	$A_{5cm}\uparrow$	$D_{corr}\downarrow$	$D_{chamf}\downarrow$	$D_{nocs}\downarrow$	$A_{5cm}\uparrow$	$A_{10cm}\uparrow$	$D_{corr}\downarrow$	$D_{chamf}\downarrow$	$D_{nocs}\downarrow$
Shirt	Ours	**29.0%**	**85.9%**	**3.88**	**1.18**	0.052	**25.4%**	**81.6%**	**8.94**	**1.85**	**0.109**
	Ours w.o. PC refiner	4.1%	25.3%	7.76	1.70	0.115	0.6%	27.4%	17.42	2.89	0.229
	Ours w.o. Mesh refiner	26.8%	83.6%	3.92	1.19	**0.048**	23.5%	81.2%	9.18	1.88	0.110
	Ours w. PointNet++ [33]	2.2%	34.7%	6.53	1.54	0.085	13.3%	53.2%	14.21	1.93	0.174
Pants	Ours	**42.8%**	**93.6%**	**3.35**	**1.10**	**0.039**	**30.7%**	**76.9%**	**9.55**	**2.71**	0.143
	Ours w.o. PC refiner	23.1%	70.0%	4.84	1.30	0.072	5.8%	49.2%	13.72	3.46	0.165
	Ours w.o. Mesh refiner	33.5%	92.2%	3.52	1.18	0.039	22.5%	75.2%	9.76	2.78	0.148
	Ours w. PointNet++ [33]	8.9%	73.6%	4.91	1.33	0.066	8.0%	69.1%	10.12	2.81	**0.125**

Table 2. Results of ablative experiments on VR-Folding dataset.

global offset o_{pc}, and Gaussian noise standard deviation δ. Besides, we also augment the canonical mesh with global scaling factor s_{mesh}. As shown in Fig. 6 (left), We can see that our method performs well at *surface reconstruction* (*i.e*. D_{chamf}) under high noise level. Please see the supplementary files for details of the noise parameters.

Robustness under Large Frame Interval Tracking at large frame intervals with huge object deformations can be very challenging. Therefore, we uniformly drop frames in videos and only keep part (*i.e*. 1/2, 1/4, 1/6, and 1/8) of the frames. The results are shown in Fig. 6 (right), we can see that our method is more robust against missing frames on *Folding* task compared to *Flattening* task.

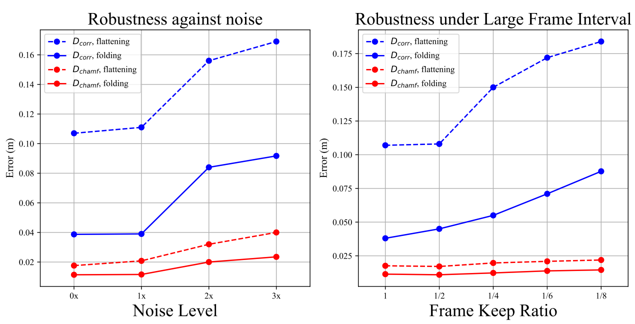

Figure 6. The robustness experiments.

5.3.4 Tracking Speed

On a single RTX 3090 GPU, GarmentNets takes 100ms to pass the backbone, 7ms for volume query, and 170ms for Marching Cubes [25]. It results in a runtime of 3.6 FPS. While in our design, we adopt a faster backbone that costs only 45ms and eliminates the time-consuming Marching Cubes. Our NOCS refiner and warp field prediction take 12ms and 7ms respectively. Our method can achieve 15 FPS during inference, which is $\sim$ 5 times faster than [11].

5.3.5 Generalization Ability

Neural Prediction as First-Frame Pose In order to evaluate the generalization ability of our method, we directly use GarmentNets prediction (*i.e*. canonical coordinates and mesh) as the first-frame pose during inference. As shown in Tab. 3, our method still outperforms GarmentNets by a large margin without any data augmentation related to GarmentNets during training.

Type	Method	$A_{3cm}\uparrow$	$A_{5cm}\uparrow$	$D_{corr}\downarrow$	$D_{chamf}\downarrow$	$D_{nocs}\downarrow$
Shirt	Ours*	**25.4%**	**78.9%**	**4.04**	**1.18**	**0.052**
	GarmentNets	0.8%	21.5%	6.40	1.58	0.221
Pants	Ours*	**45.1%**	**92.2%**	**3.33**	**1.16**	**0.040**
	GarmentNets	16.2%	69.5%	4.43	1.30	0.162
Top	Ours*	**21.1%**	**61.9%**	**4.82**	**1.11**	**0.065**
	GarmentNets	10.3%	53.8%	5.19	1.51	0.148
Skirt	Ours*	**14.7%**	**65.9%**	**5.36**	**1.46**	**0.078**
	GarmentNets	1.1%	30.3%	6.95	1.89	0.239

Table 3. Results of our method using GarmentNets prediction as first-frame pose on *Folding* task.

Real World Experiments We collect some real-world RGB-D videos of garment manipulation with Realsense L515 [5] LiDAR cameras. Our method can directly track garment pose for novel garments in the real-world with a model trained only on our simulated data. Please see the supplementary files for more qualitative results.

6. Conclusion and Future Works

In this work, we propose a complete framework for garment pose tracking, including the data collection (*i.e*. VR-Garment system), dataset (*i.e*. VR-Folding), and a strong approach (*i.e*. GarmentTracking) which is both quantitatively and qualitatively better than the baseline approach. As a platform, we believe VR-Garment can innovate the dataset collection for other kinds of deformable objects. As a manipulation dataset, we are interested in using VR-Folding for robot imitation learning. As a strong baseline, we hope GarmentTracking can facilitate future research in this challenging direction.

Acknowledgement

This work was supported by the National Key Research and Development Project of China (2021ZD0110704), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), Shanghai Qi Zhi Institute, Shanghai Science and Technology Commission (21511101200) and OpenBayes.

References

[1] HTC Vive Pro Headset. https://www.vive.com/us/product/vive-pro/. 3

[2] Noitom Hi5 VR Glove. https://www.noitom.com/hi5-vr-glove. 3

[3] Obi: Unified particle physics for Unity. http://obi.virtualmethodstudio.com/. 3

[4] Pytorch. https://pytorch.org/. 6

[5] RealSense L515 LiDAR camera. https://www.intelrealsense.com/lidar-camera-l515/. 8

[6] Unity Game Engine. http://www.unity.com/. 2, 3

[7] Yahav Avigal, Lars Berscheid, Tamim Asfour, Torsten Kröger, and Ken Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2022. 1

[8] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: clothed 3d humans. In *European Conference on Computer Vision*, pages 344–359. Springer, 2020. 1, 3, 4

[9] Aljaz Bozic, Pablo Palafox, Michael Zollhöfer, Angela Dai, Justus Thies, and Matthias Nießner. Neural non-rigid tracking. *Advances in Neural Information Processing Systems*, 33:18727–18737, 2020. 1, 2

[10] Aljaz Bozic, Michael Zollhofer, Christian Theobalt, and Matthias Nießner. Deepdeform: Learning non-rigid rgb-d reconstruction with semi-supervised data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7002–7012, 2020. 1, 2, 3

[11] Cheng Chi and Shuran Song. Garmentnets: Category-level pose estimation for garments via canonical space shape completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3324–3333, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[12] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019. 4, 7

[13] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 5

[14] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Transactions on Graphics (TOG)*, 36(6):1–16, 2017. 1, 3

[15] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 1, 3

[16] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 1, 2

[17] Huy Ha and Shuran Song. Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In *Conference on Robot Learning*, pages 24–33. PMLR, 2022. 1

[18] Fangzhou Hong, Liang Pan, Zhongang Cai, and Ziwei Liu. Garment4d: Garment reconstruction from point cloud sequences. *Advances in Neural Information Processing Systems*, 34:27940–27951, 2021. 1

[19] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. In *European conference on computer vision*, pages 362–379. Springer, 2016. 1, 2

[20] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020. 2

[21] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12706–12716, 2021. 1, 3

[22] Wenbin Lin, Chengwei Zheng, Jun-Hai Yong, and Feng Xu. Occlusionfusion: Occlusion-aware motion estimation for real-time dynamic 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1736–1745, 2022. 1, 3

[23] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. Akb-48: a real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14809–14818, 2022. 2

[24] Liu Liu, Han Xue, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Toward real-world category-level articulation pose estimation. *IEEE Transactions on Image Processing*, 31:1072–1083, 2022. 2

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 2, 8

[26] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 1, 3

[27] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 1, 2

[28] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5379–5389, 2019. 1, 3

[29] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Npms: Neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12695–12705, 2021. 1, 3

[30] Pablo Palafox, Nikolaos Sarafianos, Tony Tung, and Angela Dai. Spams: Structured implicit parametric models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12851–12860, 2022. 1, 3

[31] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. Clothcap: Seamless 4d clothing capture and retargeting. *ACM Transactions on Graphics (ToG)*, 36(4):1–15, 2017. 1

[32] 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. 5

[33] 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. 7, 8

[34] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017. 1, 2

[35] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2646–2655, 2018. 1, 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4, 5

[37] Andreas Verleysen, Matthijs Biondina, and Francis Wyffels. Video dataset of human demonstrations of folding clothing for robotic folding. *The International Journal of Robotics Research*, 39(9):1031–1036, 2020. 3

[38] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2

[39] Han Xue, Liu Liu, Wenqiang Xu, Haoyuan Fu, and Cewu Lu. Omad: Object model with articulated deformations for pose estimation and retrieval. *arXiv preprint arXiv:2112.07334*, 2021. 2

[40] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, and Shijian Lu. Pttr: Relational 3d point cloud object tracking with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8531–8540, 2022. 4

[41] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *European Conference on Computer Vision*, pages 512–530. Springer, 2020. 1, 3