# Learning Cloth Folding Tasks with Refined Flow Based Spatio-Temporal Graphs

Peng Zhou, Omar Zahra, Anqing Duan, Shengzeng Huo, Zeyu Wu and David Navarro-Alarcon

***Abstract*—Cloth folding is a widespread domestic task that is seemingly performed by humans but which is highly challenging for autonomous robots to execute due to the highly deformable nature of textiles; It is hard to engineer and learn manipulation pipelines to efficiently execute it. In this paper, we propose a new solution for robotic cloth folding (using a standard folding board) via learning from demonstrations. Our demonstration video encoding is based on a high-level abstraction, namely, a refined optical flow-based spatiotemporal graph, as opposed to a low-level encoding such as image pixels. By constructing a new spatiotemporal graph with an advanced visual corresponding descriptor, the policy learning can focus on key points and relations with a 3D spatial configuration, which allows to quickly generalize across different environments. To further boost the policy searching, we combine optical flow and static motion saliency maps to discriminate the dominant motions for better handling the system dynamics in real-time, which aligns with the attentional motion mechanism that dominates the human imitation process. To validate the proposed approach, we analyze the manual folding procedure and developed a custom-made end-effector to efficiently interact with the folding board. Multiple experiments on a real robotic platform were conducted to validate the effectiveness and robustness of the proposed method.**

***Index Terms*—Robotic Manipulation; Cloth Folding; Learning from Demonstration; Spatiotemporal Graph; Robot Vision.**

## I. Introduction

ROBOTS have become a crucial part of advancing the manufacturing industry in the past few decades and are widely used in domestic environments nowadays [1]. Cloth folding is high on the list of monotonous home duties that many people dislike and could, theoretically, be performed by a service robot [2], which seems to be simple tasks for a human, yet, it poses significant challenges for a robotic system. Most industrial robots were designed to do repetitive tasks with rigid objects. Clothes, however, require several additional skills currently unavailable to robots [3].

In contrast to their rigid counterpart, deformable objects (e.g., fabric) may deform substantially and form wrinkles or folds [4], significantly increasing the difficulty of its manipulation. [5] and [6] perform successful manipulation tasks on elastic objects by estimating a deformation model from vision and motion sensory feedback. However, creasing of clothes upon force exertion complicates the process of modelling and prediction, and complicates the visual perception of the cloth state as well. Thus, research efforts [7]–[9] have been focused on this complex state representation and estimation.

This work is supported in part by the Research Grants Council of Hong Kong under grants 14203917 and 15212721, in part by the Key-Area Research and Development Program of Guangdong Province 2020 under project 76, and in part by the Jiangsu Industrial Technology Research Institute Collaborative Research Program Scheme under grant ZG9V.

All authors are with The Hong Kong Polytechnic University, Department of Mechanical Engineering, Hung Hom, KLN, Hong Kong.

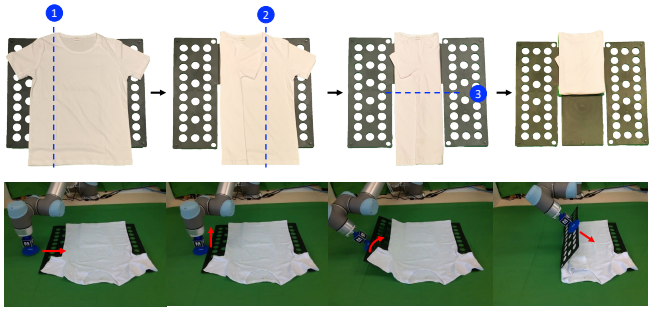

Fig. 1. (Up) The folding process with folding board can be divided into *Left folding*, *Right folding* and *Mid-folding*. (Down) Each robotic folding process can be decomposed into *Approaching*, *Lifting*, *Rotating* and *Pushing* steps.

To tackle these challenges, we propose a new approach to perform the cloth folding task by using a standard manual-folding board, as shown in Fig. 5(c). Instead of designing a complex manipulator [10] or a versatile gripper [11] to directly have contact with clothes based on real-time measurements on the highly deformable textiles, we aim to solve clothes folding by using this assistive tool to construct limited and unified manipulation primitives, which allows us to only estimate few intermediate states, instead of measuring deformations frequently. Fig. 1 shows a folding process by using a folding board, and the blue dashed lines represent the folding line of each operation. According to the common practice in daily life and benchmarks for cloth manipulation [12], we divide the folding task into three sub-tasks, namely, i) *left folding*; ii) *right folding*; and iii) *mid-folding*. Subsequently, each sub-folding could be partitioned into four processes, namely, i) *approaching*; ii) *lifting*; iii) *rotating* and iv) *pushing* (see details in Sec. IV).

On the other hand, the ability of Learning from Demonstration (LfD) [13] - called imitation learning [14] or third-person imitation learning [15] - has long been a desirable objective in artificial intelligence as a method to train agents intuitively rather than hard-coding their actions rapidly. For visual imitation to be successful, the demonstrator's sensory environment must be well understood, including how it changes over time [16]. Visual imitation is thus reduced to learning a visual similarity function between the demonstration and imitation scenes, which might be maximized by the imitated behaviors, resulting in accurate skill imitator behavior. This similarity function establishes which elements of the visual observations are essential to replicate the demonstrated behaviors. Put

differently, and it determines what to imitate and what to disregard [17]. Consequently, we formulate the cloth folding task as an imitation learning problem using an assistive folding board. In detail, the robot agent needs to search a policy to reproduce the successful folding task from video sequences of presented demonstrations by a human expert.

Imitation learning tackles the issue of skill acquisition via demonstration observation [18]. However, most prior methods [19], [20] presuppose that demonstrations are provided in the agent's workplace. Typically, a mapping between demonstrator and imitator spatial observations is needed and is critical for effective imitation [21]. Direct comparison of pixel intensities, on the other hand, is not a reliable indicator of resemblance since it may be tainted by the difference in viewpoints [22], lighting changes [23], or changed poses [24]. Recent method [25] has learned such visual similarity by simply training and matching entire image feature embeddings and bypass the specific identification of the environment structure in terms of detected objects' positions and orientations. Instead, our method tries to capture a high-level abstraction input first and then train a policy to better handle spatiotermporal dynamics, which is proved to have benefits for generalization of action-conditioned dynamics. [26] has used such graph-encodings to learn a model predictive control in a real-world application.

To solve imitation learning for cloth folding tasks, we propose a hierarchical encoding for dominant motion representations extracted from a single demonstration video, called a refined optical flow-based spatiotemporal graph (STG), where vertices represent corresponding key points of the manipulated objects (i.e., the folding board) tracked throughout space and time, and edges denote their relative 3D spatial configurations. Our demonstration video encoding is based on a high-level abstraction, a refined optical flow-based STG, instead of a low-level encoding (i.e., image pixels). For each pair of time steps, we build two refined STGs (see Fig. 2d), one for the demonstration workspace and one for the imitation. Then, our reward function measures the dissimilarity between corresponding 3D relational vertex pairs and optimizes the policy updated with reinforcement learning algorithms from a single cloth folding demonstration video. To validate the effectiveness of the proposed approach, we conducted multiple experimental studies on a real robotic platform. In summary, the original contributions of this work are as follows:

- A new approach for cloth folding task based on an assistive folding board to avoid complex measurements of the highly deformable clothes.
- A refined optical flow-based STG designed for detecting dominant motion fields with the aim of accelerating the valid policy searching.
- A hierarchical abstraction (i.e., STG) for demonstration video encoding.

Although this study involves solving the problem of clothes folding, the proposed methods target a broad set of manipulation problems. Specifically, it targets problems involving manipulation of non-rigid objects through imitation while considering both spatial and temporal graph-structured cues demonstrated by the teaching agent.

The rest of this letter is organized as follows: Sec. II formulates the problem; Sec. III presents the proposed approach; Sec. IV reports the experiments; Sec. V gives final conclusions. A video of the conducted experiments can be accessed at: https://sites.google.com/view/learnfolding.

## II. Formulation

We define two spatio-temporal graph sequences $\mathcal{G}_t^E = \{\mathcal{V}_t^E, \mathcal{E}_t^E \mid t = 1, 2, \cdots, N\}$ and $\mathcal{G}_t^R = \{\mathcal{V}_t^R, \mathcal{E}_t^R \mid t = 1, 2, \cdots, N\}$ extracted from a demonstration video of a human expert and an imitation video of a robotic agent, respectively, in the same time sequence length denoted by $N$. This high-level abstraction aligns with our human attentional mechanism during the learning process. Therefore, this built graph is independent of its workspace as long as it can be encoded from the observation successfully. Formally, let $\mathcal{V}_t = \left\{v_t^1, \cdots, v_t^n \mid v_t^i \in \mathbb{R}^3\right\}$ be the set of vertex of the corresponding spatio-temporal graph at the $t$ time step and $v_t^i$ represents the $i$-th visual corresponding point during the manipulation process. A vertex $v_t^i$ could be an object's position, object part's position, or any interest point detected from the observation during the demonstration or imitation process. Subsequently, $\mathcal{E}_t = \left\{e_t^1, \cdots, e_t^m \mid e_t^j \in \{v_t^j, v_t^{j+1}\}\right\} \subseteq \mathcal{V}_t \times \mathcal{V}_t$ represents the edge set and $e_t^j$ denotes the $j$-th spatial edge to be maintained during a manipulation process between corresponding vertices, $v_t^j$ and $v_t^{j+1}$. As shown in Fig. 2, at the same time step $t$, the graph of the expert's demonstration $\mathcal{G}_t^{Expert}$ and the one of the robot agent $\mathcal{G}_t^{Robot}$ have the same structure, which means same cardinal number for the corresponding set, $\mathbf{card}(\mathcal{V}_t^E) = \mathbf{card}(\mathcal{V}_t^R)$, $\mathbf{card}(\mathcal{E}_t^E) = \mathbf{card}(\mathcal{E}_t^R)$ and same vertex pair for any edge, $e_t^{E,j} = e_t^{R,j} \mid \forall e_t^{E,j} \in \mathcal{V}_t^E, e_t^{R,j} \in \mathcal{V}_t^R$. During the entire manipulation, a graph vertex $v_t$ and a graph edge $e_t$ could be added into or deleted from the spatio-temporal graph with the manipulated object changing. We only ensure that the graph built from the expert and the one of robot agent possess a same structure at the same time step.

In the context of cloth folding, we only consider three types of vertices: folding board vertices $\mathcal{V}_b$, cloth vertices $\mathcal{V}_c$, end-effector vertices $\mathcal{V}_e$. Folding board vertices are detected or inferred key points to describe its skeleton during the folding. Cloth vertices describe the real-time shape (i.e., contour, surface, mesh) during the folding process. Last, the end-effector vertex represents the pose of end-effector bottom which contacts with the folding board. Structural dissimilarity at a time step $t$ between the human expert graph $\mathcal{G}_t^E$ and robot agent graph $\mathcal{G}_t^R$ is calculated as our loss function as below:

$$\mathcal{L}(\mathcal{G}_t^E, \mathcal{G}_t^R) = \sum_{i \in \mathcal{V}_e, j \in \mathcal{V}_b} w(\mathcal{E}_t^{(i,j)}) \cdot F(\mathcal{E}_t^{(i,j)} \wedge f(\mathcal{E}_t^{(i,j)})) \cdot \left\| (v_t^{E,i} - v_t^{E,j}) - (v_t^{R,i} - v_t^{R,j}) \right\| \quad (1)$$

where $w(\mathcal{E}_t^{(i,j)}) \in \mathbb{R}$ is the weight of the corresponding edge $\mathcal{E}_t^{(i,j)}$ between end-effector vertex set $\mathcal{V}_e$ and folding board vertex set $\mathcal{V}_b$ during imitating clothes folding, and $F(\mathcal{E}_t^{(i,j)} \wedge f(\mathcal{E}_t^{(i,j)})) \in \{0, 1\}$ is a Boolean function of the edge $\mathcal{E}_t^{(i,j)}$ and its optical flow function $f(\mathcal{E}_t^{(i,j)})$ to indicate whether $\mathcal{E}_t^{(i,j)}$ is preserved according to current sequence of the refined optical

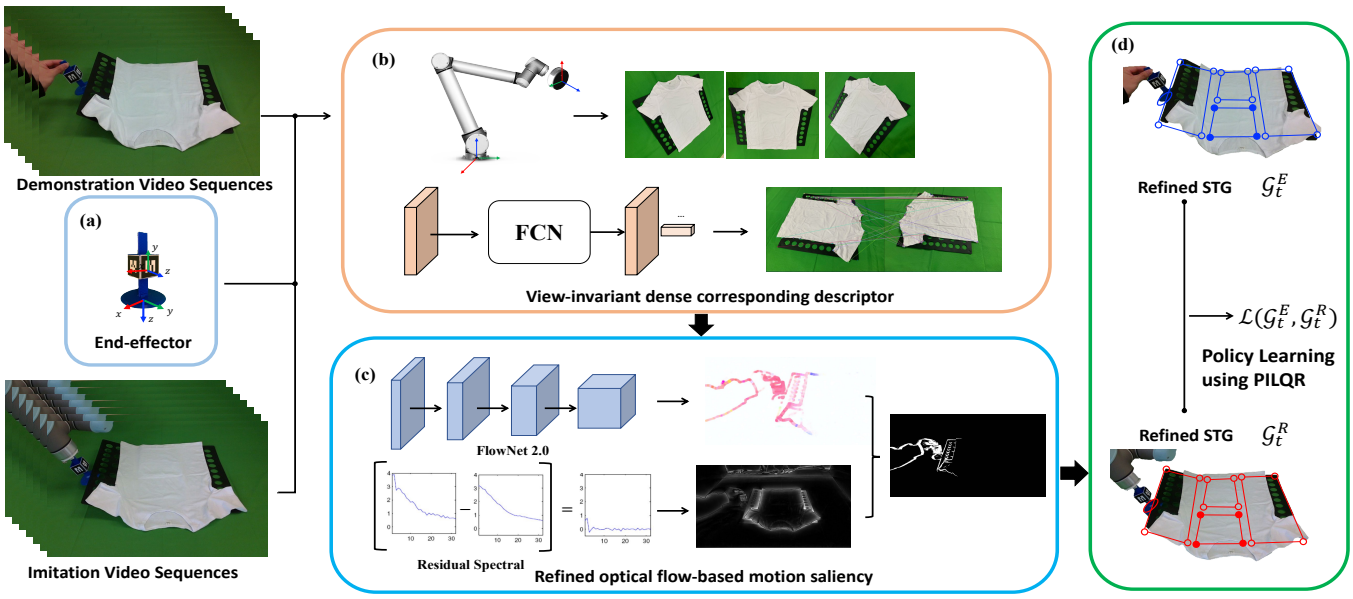


Fig. 2. Conceptual representation of the proposed approach for cloth folding tasks, which encodes the video sequences into a refined optical flow-based STG across the demonstrator's and imitator's workspace. A well-designed end-effector and a view-invariant dense corresponding descriptor can form the basic structure of an STG, which will be refined with the optical flow-based motion saliency map to optimize the corresponding folding policy.

flow. We regard weight $w(\mathcal{E}_t^{(i,j)})$ as hyperparameters for our framework, and at present we set them with empirical values. As for learning optimal values of these weights, we put this important task in our future work.

## III. Methodology

### A. Spatio-Temporal Graph (STG) Construction

To robustly extract a high-level abstraction for each video frame, we decompose the spatio-temporal graph into three modularized components, namely, 1) *End-effector detection*: the pose of the end-effector must be robustly detected in a real-time manner across the demonstration or imitation workspaces; 2) *Folding board detection*: the graph-structured key points (see Fig. 3 (a)) will be extracted partially from the original observation and inferred partially from their prior knowledge (i.e., 3D spatial relationships). 3) *Clothing state estimation*: the state of the clothing must be measured after each flipping operation so that we can identify whether the task is solved successfully.

*1) End-effector Design and Detection:* By analyzing the procedures for each flipping, it is hard to estimate the end-effector's pose at a real-time rate, because sometimes the end-effector could be obscured by the folding board when the end-effector is having contact with the folding board. As shown in Fig. 5 (b), we design and 3D-print modularized components, which includes a bottom circle plate to contact with the folder, a cube with four-sided AR markers to detect its pose, and a long strip to connect the above components. Particularly, there is a slope between the upper and lower surfaces of the bottom plate to allow sliding below the folder easily while approaching. In addition, a circular shape design can ensure sufficient and robust contact during lifting, rotating and pushing. Besides, we calibrate and fix the transformation from the center of the bottom circle plate to each AR marker. With this transformation, we can robustly detect the pose of the bottom circle plate over demonstration and imitation workplaces when the folding board obscures the end-effector with close contact in a single camera perspective.

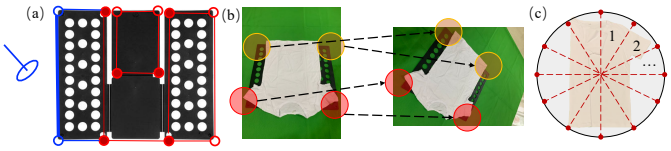


Fig. 3. Conceptual representation of STG extractions and clothes state estimation.

*2) Folding Board Detection:* We consider folding board detection as a view-invariant correspondence and spatial structure inference problem. Although using a folder to assist in folding clothes can avoid the complicated real-time measurement of the highly deformable object, obtaining a precise 3D spatial structure of the folder partially overlapped by the clothes is not an easy task. To alleviate such difficulty, we take the spatial relationships among the key points at the static scene as prior knowledge to infer the entire folder structure. Precisely, we first extract four corners (see Fig. 3b) with Harris cornet detector at the initial state. In case no motions of the folding board occur when the end-effector is approaching the first contact point, we define the edge that will move in the nearest future as an anchor edge. In contrast, the rest vertices could be calculated according to the initially defined fixed spatial relationships. The most challenging folding subtask is the mid-folding where the left and right folding often cover the upper corner points of the mid-section of the folding board. For this case, we calculate the upper 3D positions based on an average normal of the clothing surface and a fixed spatial

transformation defined in the initial state.

With the four corner points, we leverage self-generated visual correspondences to build a powerful dense feature descriptor, which aims at establishing their correlations across the demonstrator and imitator's working place. To formalize, let $\mathcal{I}$ denote an image with a $h \times w$ resolution, and $\mathcal{D}$ a non-linear point descriptor without modifying its resolution. We are targeting on learning a densely $d$-dimensional descriptor $\mathcal{D}(\mathcal{I}) : \mathbb{R}^{h\times w\times 3} \rightarrow \mathbb{R}^{h\times w\times d}$ to encode the identity for each pixel. In order to accelerate the training process and optimize computing resource utilization for our dense descriptor, we employed the backbone of Fully Convolutional Networks [27], which could reuse the computed activations for overlapping pixels, from images with arbitrary resolution, and extract the dense descriptor efficiently. During the network training process, assume for each flipping we have a video clip $\mathcal{V} = \{(\mathcal{I}_1, \mathcal{M}_1), \cdots, (\mathcal{I}_N, \mathcal{M}_N)\}$ of $N$ frames, where $\mathcal{M}_n$ corresponds to a universal correspondence model of an image $\mathcal{I}_n$ to provide a mapping from pixels in $\mathcal{I}_n$ to the correspondence coordinates in other frames. Only the relationship between different correspondence coordinates in local reference frames are preserved. Consequently, we adopt a strategy of pairwise correspondence coordinate labeling and take a pairwise contrastive loss [27] in pixel-level defined as:

$$L(\mathcal{D}(\mathcal{I}_{\mathbf{x}}), \mathcal{D}(\mathcal{I}'_{\mathbf{x}'}), \mathcal{M}(\mathcal{I}_{\mathbf{x}}), \mathcal{M}(\mathcal{I}'_{\mathbf{x}'})) = \begin{cases} \left\|\mathcal{D}\left(\mathcal{I}_{\mathbf{x}}\right) - \mathcal{D}\left(\mathcal{I}'_{\mathbf{x}'}\right)\right\|^2 & \mathcal{M}(\mathcal{I}_{\mathbf{x}}) = \mathcal{M}(\mathcal{I}'_{\mathbf{x}'}) \\ \max\left(0, \xi - \left\|\mathcal{D}(\mathcal{I}_{\mathbf{x}}) - \mathcal{D}(\mathcal{I}'_{\mathbf{x}'})\right\|\right)^2 & \text{Otherwise} \end{cases} \tag{2}$$

where $\mathcal{D}(\mathcal{I}_{\mathbf{x}})$ and $\mathcal{D}(\mathcal{I}'_{\mathbf{x}'})$ denote the encoded descriptor for image $\mathcal{I}$ at coordinate $\mathbf{x} = (x, y)$ and image $\mathcal{I}'$ at coordinate $\mathbf{x}' = (x', y')$, respectively. If the correspondence model of coordinates $\mathbf{x}$ and $\mathbf{x}'$ can be mapped to the same 3D point, we regard them as a positive pixel pair to minimize the distance in the feature space built with the correspondence descriptor. Otherwise, the contrastive loss will split them at least $\xi$ margin away. We automate the collection of multiview image sequences of the robotic agent's workspace by using an RGB-D camera mounted to the robot's end-effector and moving the camera along random paths that cover a variety of viewpoints of the scene and varying distances from the objects. We estimate the camera pose via hand-eye calibration using the robot's forward kinematics model, which, when combined with known intrinsic parameters and aligned depth pictures, enables robust 3D reconstruction of the scene and correct pixel correspondences across multiple viewpoints.

*3) Clothing State Estimation:* Clothing state identification will be executed after each folding task. To robustly estimate the state from multiple perspectives, we build a simple but efficient classifier by combining radial region-based features and view-invariant moments features. As shown in Fig. 3c, we divide the mask of the extracted clothing into 12 radial bins after computing the centroid as below:

$$C_m = \left(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}\right) \tag{3}$$

For each radial area, we compute a feature vector formed by averaging critical points, area, eccentricity, perimeter and orientation similarly defined in [28]. Based on Hu-Moments invariant features [29], the moment $(p, q)$ of an image $f(x, y)$ of size $M \times N$ is defined as: $m_{p,q} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) x^p y^q$, where $p$ and $q$ are the order of $x$ and $y$, respectively. We compute the clothing moment similarly, except that $x$ and $y$ are displaced by the mean values as follows:

$$\mu_{p,q} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \left(x - \bar{x}\right)^p \left(y - \bar{y}\right)^q \tag{4}$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$ and $\bar{y} = \frac{m_{01}}{m_{00}}$

By applying normalization, clothing moments are defined as follows:

$$\eta_{\mathrm{p,q}} = \frac{\mu_{\mathrm{p,q}}}{\mu 00^{\gamma}}, \gamma = \frac{\mathrm{p} + \mathrm{q} + 2}{2}, \mathrm{p} + \mathrm{q} = 2, 3, \ldots \tag{5}$$

Finally, based on these clothing moments, 7-dimensional Hu-moments could be calculated. By concatenating two group features, a Support Vector Machine (SVM) is implemented to identify whether a clothing state is a correct one after the corresponding folding.

### *B. Refined Optical Flow-based STG*

After completing the STG construction, we propose extracting the refined optical flow from the object saliency map to determine which edge must be preserved during the policy learning process. We anticipate that the optical flow field should be sufficiently identifiable in prominent regions. To this end, we will compare the flow field with a real-time static saliency map which would have enclosed the calculated optical flow in the identical regions. The former may be determined using an optical flow technique. The latter is not immediately accessible because it is not noticed. Nonetheless, it is predictable using a refined optical flow technique. This is precisely the novelty of utilizing refined flow to denote real-time motion saliency to identify the presence of STG edges.

Our overall framework is illustrated in Fig. 4. Given two adjacent frames $\mathcal{I}_t$ and $\mathcal{I}_{t-1}$ at time step $t$ and $t-1$ in the video clippings, we first calculate its real-time optical flow $\mathcal{F}_t$ with FlowNet 2.0 [30]. However, the motion boundary generated with FlowNet 2.0 lacks details in few small but important regions. Therefore, we compute another object saliency map with Spectral Residual (SR) algorithm [31] $\mathcal{S}_t$ to paint the raw optical flow in order to yield a refined optical flow to indicate the motion saliency in a real-time manner. For the motion contour, we could directly employ a threshold on the norm of the gradient of the velocity vectors. However, this method will yield noisy contours. Instead, we choose to perform the classical Canny [32] edge detection method on the transformed image with an HSV color scheme. Then the region enclosed by motion contour $\mathcal{C}_t$ generated with respect to the raw optical flow $\mathcal{F}_t$ is regarded as an inpainted mask $\mathcal{C}_t$ to cut the static saliency map $\mathcal{S}_t$. Finally, a threshold is performed to yield the final refined flow $\Re_t$.

The refined optical flow, i.e., the object saliency map $\mathcal{S}_t$ located inside the motion contour $\mathcal{C}_t$ enclosed by the optical

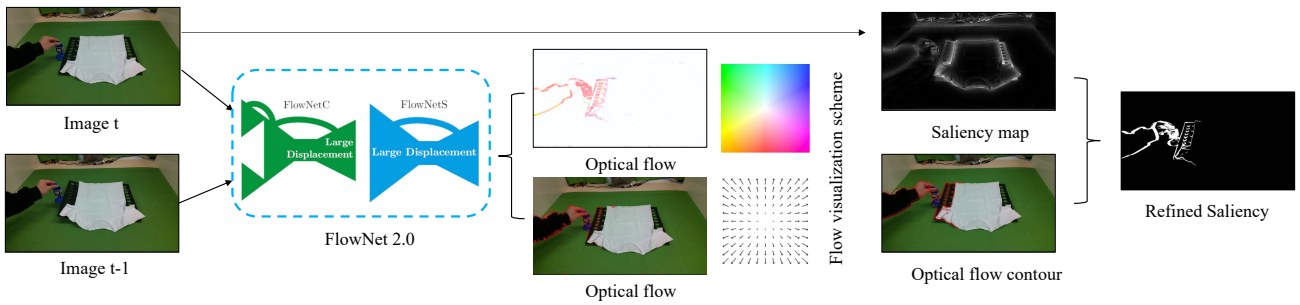


Fig. 4. The flow chart of the refined optical flow-based motion saliency map. Given an image pair at time step $t$ and $t-1$, $\mathcal{I}_t$ and $\mathcal{I}_{t-1}$, an optical flow $\mathcal{F}_t$ can be detected with FlowNet 2.0 and then be refined based on the motion contour $\mathcal{C}_t$ extracted from a static motion saliency $\mathcal{S}_t$.

flow $\mathcal{F}_t$, is computed over the inpainted mask indicated by the motion contour. From the refined flow, a motion saliency map is expected to be defined within $[0, 1]$ as follows:

$$\forall \mathbf{x} \in \mathcal{C}_t, \quad \Re(\mathbf{x}) = 1 - \exp(-\lambda \|\mathcal{R}_t(\mathbf{x})\|_2) \tag{6}$$

where $\Re(\mathbf{x}) = 0$ for $\mathbf{x} \notin \mathcal{C}_t$ and $\lambda$ regulates the refined optical flow score. Therefore, $\Re(\mathbf{x})$ output a non-zero flow score highlighting the moving elements, while $\lambda$ can be regarded as a trade-off between robustness to noise and ability to highlight tiny but still salient motions. Specifically, we can set a high $\lambda$ to produce a binary map to explicitly present motion segmentation, as shown in Fig. 4(e). To this end, we introduce a parameter $\varepsilon$ to segment the refined flow as below:

$$\|\mathcal{R}_t(\mathbf{x})\|_2 \geqslant -\frac{\ln(1-\varepsilon)}{\lambda} \tag{7}$$

In this case, the refined flow magnitude at an image location $\mathbf{x}$ larger than $\frac{\ln 2}{\lambda}$ will be segmented out if $\varepsilon$ is set to 0.5. Therefore, the introduction of $\lambda$ is able to add relative flexibility into this workflow built for refined optical flow.

### C. Motion Policy with Refined Flow-based STG

We cast motion policy learning using a refined flow-based STG as a reinforcement learning problem. With the optimized policy, we are aiming at teaching the robot to learn cloth folding from a single demonstration video of human experts. Formally, for each time step $t$ of the cloth folding task, the $\theta$-parameterized policy $\pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)$ identifies a probability distribution over actions $\mathbf{a}_t$ constrained by current system state $\mathbf{s}_t$. Let $\tau = (\mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T)$ denote a trajectory of states and actions of cloth folding. With a loss function $l(\mathbf{s}_t, \mathbf{a}_t)$, the entire trajectory cost can be formulated as:

$$J(\theta) = \mathbb{E}_\pi[l(\tau)] = \int l(\tau)\pi(\tau)d\tau \tag{8}$$

where $\pi(\tau)$ is the distribution of policy trajectory under the system dynamics $\pi(\mathbf{s}_{t+1} \mid \mathbf{x}_t, \mathbf{a}_t)$ defined as below:

$$\pi(\tau) = \pi(\mathbf{s}_1) \prod_{t=1}^{T} \pi(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\, \pi(\mathbf{a}_t \mid \mathbf{s}_t) \tag{9}$$

The action $\mathbf{a}_t$ is defined as relative pose alteration $\Delta\mathbf{p}_t = \mathbf{p}_t - \mathbf{p}_{t-1}$ of the robot end-effector, which is composed of a 7-dimensional vector $(\Delta x, \Delta y, \Delta z, \Delta q_1, \Delta q_2, \Delta q_3, \Delta q_4)$, where $\Delta x, \Delta y, \Delta z$ represent the position's shift and $\Delta q_1, \Delta q_2, \Delta q_3, \Delta q_4)$ denotes the orientation's shift. The state $\mathbf{s}_t$ is defined as a vector consisting of end-effector's 3D positions, robot joint angles and graph spatial configurations of the workspace. For $n$ detected anchor points, this vectorization scheme results in a $3 + n_{\text{joints}} + n_{\text{anchor\_points}} * 3 + d_{\text{anchor\_points}}$ dimensional state space, where $n_{\text{joints}}$ is the number of robot joints, $n_{\text{anchor\_points}}$ denotes the anchor points detected from the active section of the folding board, and $d_{\text{anchor\_points}}$ indicates the distances between the end-effector and different anchor points. In our case, the distance between different anchor points are ignored because their spatial relationships are relatively fixed on the folding board. While for imitation learning with multiple objects, it is still needed to preserve.

To take the advantages of model-free and model-based reinforcement learning approaches at the same time, a state-of-the-art technique called PILQR [33] is applied to minimize the entire trajectory cost defined in Eq. (1). This approach optimizes TVLG policies by combining rapid model-based updates via iterative linear-Gaussian model fitting and improved model-free updates in the PI$^2$ framework. Consequently, it is capable of combining model-based learning's efficiency with the generality of model-free updates with the aim at solving complicated continuous control problems, which are infeasible to perform solely using either linear-Gaussian models or PI$^2$ alone. Besides, it is able to maintain orders of magnitude more efficient than conventional model-free RL. Finally, a time-dependent policy is learned as below:

$$\pi_t(\mathbf{a}_t \mid \mathbf{s}_t; \theta) = \mathcal{N}(\mathbf{K}_t\mathbf{s}_t + \mathbf{k}_t, \mathbf{\Sigma}_t) \tag{10}$$

where the time-dependent control gains are optimized by alternating model-based and model-free updates. This way, the dynamic model $p(\mathbf{s}_t \mid \mathbf{s}_{t-1}, \mathbf{a}_t)$ is learned during training time.

## IV. Results

In this section, the experimental setup for learning cloth folding is described first, and afterward, the learned policy with our approach is compared with the one with a basic STG and other existing approaches. Three tasks performed on a T-shirt are considered to imitate: **left folding**, **right folding** and **mid-folding**. For each folding, we define four procedures, namely, *approaching*: the end-effector is approaching the first contact position to insert the bottom circle board between the left side and workbench; *lifting*: the end-effector is lifting up

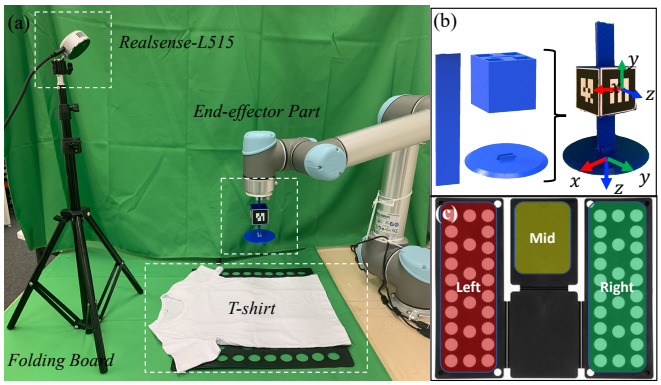


Fig. 5. (a) shows the experimental setup for learning cloth folding task using an assistive folding board; (b) presents the designed end-effector for a robust pose estimation; (c) gives the top view of a folding board and different color represents different folding section.

to obtain enough space for rotating; *rotating*: the end-effector is rotated to align with the folding board plane; *pushing*: the end-effector is pushing the corresponding folder section with a certain speed to finish the task. To evaluate the generalizability, we also extend the approach to the shorts folding task.

### A. Experiment Setup

To validate the effectiveness of the policy learned via refined optical flow-based STGs from a single demonstration video in a robotic cloth folding task using a folding board, a UR-5 robot is utilized to perform the different folding tasks with a well-designed end-effector. As shown in Fig. 5, in order to robustly detect the real-time pose of the end-effector, we design and 3D-print modularized components as shown in Fig. 5, which includes a bottom circle plate to contact with the folder, a cube with four-sided AR markers to detect its pose, and a long strip to connect the above components. We calibrate and fix the transformation from the center of the bottom circle plate to each AR marker. With this transformation, we can robustly detect the pose of the bottom circle plate over demonstration and imitation workplaces when the folding board obscures the end-effector with close contact in a single camera perspective. The entire experimental setup is shown in Fig. 5, where we use an RGB-D camera (RealSense L-515) to observe the cloth folding manipulation process using a folding board (see Fig. $5(\mathrm{c})$, and the clothing (T-shirt or shorts) is selected to evaluate the generated folding policy. Note that each piece of clothing is laid out on the board smoothly and flat at the initial state.

### B. Comparison with Existing Methods

Time-contrastive networks (TCN) [34] — a self-supervised approach for robotic imitation learning — and a pure STG are selected to compare against our learning approach using the refined optical flow based-STG. TCN trains a viewpoint-invariant representation in an embedding space ($g : \mathcal{X} \rightarrow \mathcal{Z}$) with a reward function defined based on the squared Euclidean distance and a Huber-style loss, which is formulated as below:

$$R\left(\mathbf{z}_t^E, \mathbf{z}_t^R\right) = -\alpha \left\|\mathbf{z}_t^E - \mathbf{z}_t^R\right\|_2^2 - \beta\sqrt{\gamma + \left\|\mathbf{z}_t^E - \mathbf{z}_t^R\right\|_2^2} \quad (11)$$

where $\mathbf{z}_t^E$ and $\mathbf{z}_t^R$ is the TCN embedding calculated from the demonstration video sequences of human experts and the imitation video sequence of robot agents, respectively. Here, $\alpha$ and $\beta$ are weighting parameters to control policy updates magnitude and task execution precision. In our experiment, we implement the exact same network architecture designed in [34], which is composed of the Inception model pre-trained with ImageNet up to the Mixed 5d layer, connected with two convolutional layers, a spatial softmax layer, and a fully-connected layer outputting a 32-dimensional embedding representation for the original input image. Sixteen video recordings, eight human expert demonstrations, and eight robot imitation executions for each folding task are collected to train a corresponding TCN representation. In contrast, one demonstration is provided for learning the folding policy with our refined optical flow-based STG in the same system and environment configurations. During training session of TCN models, $\alpha$ and $\beta$ are set to $1.0$ and $0.001$, respectively, while $\gamma = 10^{-5}$. Though only one demonstration video is needed for our approach training, policy searching with the refined optical flow-based STG requires a view-invariant dense correspondence descriptor and creating a refined optical flow. Thus, we collected the same amount of data for training the TCN baseline for a fair comparison. To compare the proposed graph-structured encoding with previous convolutional image encoding [34], we conduct experiments to test the robustness of our method against detector failures and occlusions, as well as its generalizability across clothing of various categories and textures. We also assess the method's robustness to the changes in initial object spatial configurations.

Fig. $6(\mathrm{e})$ depicts our method's reward curves as well as the pure STG and TCN baseline for different robot imitation videos, indicating how effectively the robot imitates the human demonstration. The horizontal axis represents time, while the vertical axis represents the cost of imitation. Despite viewpoint changed in the 3rd row, the proposed refined graph-based cost function correctly detects all correct robot imitations and accurately signals the incorrect imitation segments in the 2nd row. The baseline TCN cost curves, on the other hand, are non-discriminative. Lastly, though the STG cost curve shows a relatively higher than TCN curve, it can not compete with our method. Cost curves with high discrimination are essential for successful policy learning, and details will be discussed in the following.

### C. Discussion

In this task, first, the clothing is laid out in the center of the folding board smoothly and flat. Second, according to the common practice for clothing folding, the robot needs to go through four different processes. Namely, i) Approaching ii) Lifting, iii) Rotating, and iv) Pushing. For each folding, the robot will execute the abovementioned four common processes with slight motion changes.

In order to measure the performance of the refined STG approach, we set a home position setting above the folding board with a certain distance (50 cm). For each method, we run the policy eight times, starting from the home position.

TABLE I
SUCCESS RATE OF LEFT FOLDING, RIGHT FOLDING AND MID-FOLDING TASKS ON A T-SHIRT.

| Process | Ref. STG | STG | TCN |
|---|---|---|---|
| Approaching | 8/8 | 8/8 | 3/8 |
| Lifting | 8/8 | 7/8 | 2/3 |
| Rotating | 8/8 | 6/7 | 2/2 |
| Pushing | 7/8 | 5/6 | 1/2 |
| Summary | 7/8 | 5/8 | 1/8 |

| Process | Ref. STG | STG | TCN |
|---|---|---|---|
| Approaching | 8/8 | 8/8 | 4/8 |
| Lifting | 8/8 | 8/8 | 3/4 |
| Rotating | 8/8 | 7/8 | 3/3 |
| Pushing | 8/8 | 5/7 | 1/3 |
| Summary | 8/8 | 5/8 | 1/8 |

| Process | Ref. STG | STG | TCN |
|---|---|---|---|
| Approaching | 8/8 | 8/8 | 3/8 |
| Lifting | 7/8 | 7/8 | 2/3 |
| Rotating | 7/7 | 6/7 | 1/2 |
| Pushing | 6/7 | 4/6 | 0/1 |
| Summary | 6/8 | 4/8 | 0/8 |

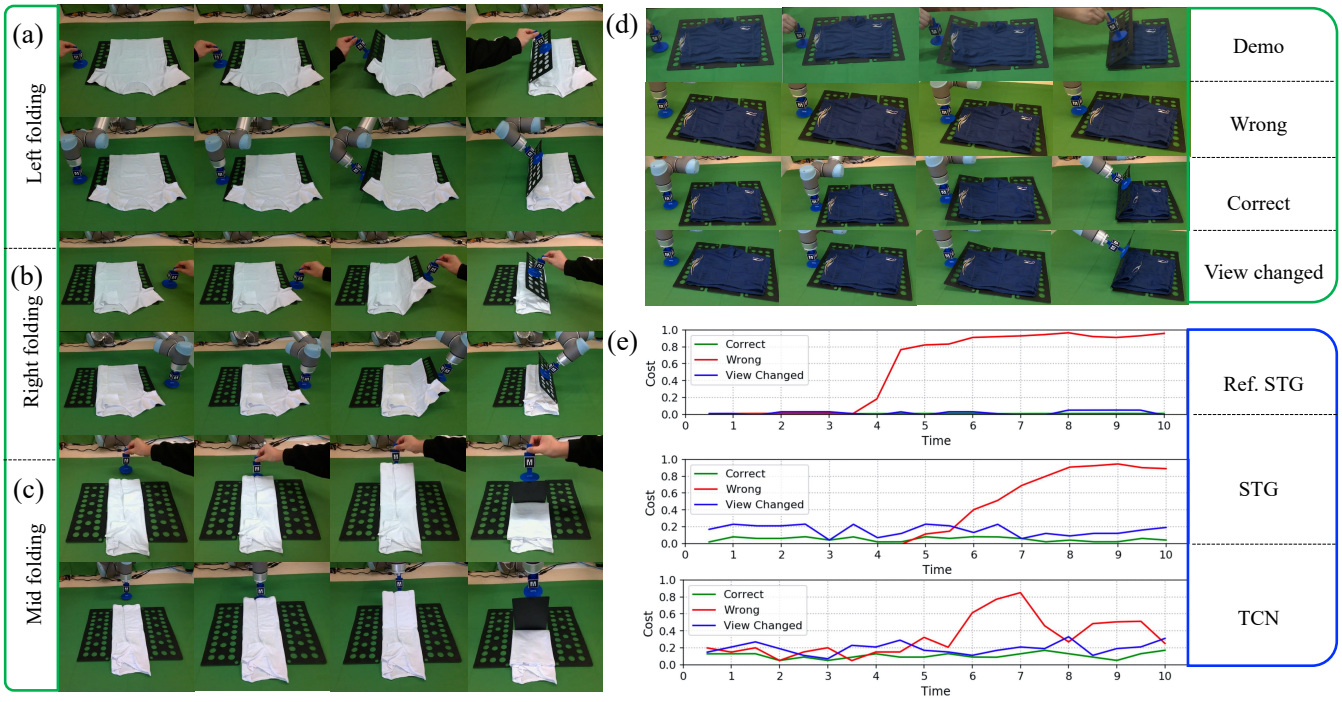


Fig. 6. Performance of the designed folding task. (a)-(c) present the successful folding examples for different folding tasks, respectively. (d) shows different robot imitation videos to compare the imitation efficiency. (e) shows the corresponding reward curves of different approaches on above imitation videos.

We consider the task success only when the four sub-processes are all solved because these four sub-processes are executed sequentially. Any unsuccessful sub-process will lead to the failure of the entire task. The detailed success rate of each process for the different folding tasks is reported in Table I. The robot successfully solves the left folding for seven runs out of 8 with our proposed approach, demonstrating a solid ability to handle complexity and dynamics. STG gets a success rate of 5 runs out of 8, which is acceptable compared with the TCN-based approach. TCN failed in almost all runs at the first sub-process. At the same time, STG performs better at the beginning of each process since graph-structured state representation can capture and handle spatiotemporal dynamics. However, it performs poorly when handling the rest folding process such as *pushing* because only the part of the related motion is vital for policy-shaping instead of the entire structure of the STG. With refined optical flow, our method can help reinforcement learning algorithms optimize the policy efficiently, focusing on system dynamics caused by the detected core motions in every policy update.

Right folding is similar to left folding because they are mirror operations in spatial configurations. Therefore, the success rate shares the same pattern with the left folding task. Specifically, our method completed this task for all runs out of 8, and one of the cases is shown in Fig. 6. On the other hand, the mid-folding is a challenging task of imitating from the demonstration videos. As shown in Fig. 6, the viewpoint of the camera is in the front of the folding board with an article of clothing after two folds. Therefore, the graph difference caused by motion changes is harder to detect compared to the other abovementioned folding tasks, which is aligned with the success rate shown in the Tab. I. From the table, we can tell that all testing approaches perform worse at mid-folding compared to the other folding subtasks. Specifically, TCN failed all the runs finally. On the other hand, STG only has a half success rate for four runs out of 8. However, our method sill can successfully complete six runs out of 8 in total. Generally, TCN stands for the policy searching purely based on the states laying in the embedding space generated from pure image inputs, which handles complex spatiotemporal dynamics poorly, while the STG-based approach handles such dynamics better. However, it lacks an attentional motion mechanism that can focus on the core structure caused by the significant motions and help

accelerate policy learning updates.

### D. Limitations

We attempt to solve clothing folding in a perspective of visual imitation using a refined optical flow-based spatiotemporal graph structure as its input. Consequently, constructing a precise refined flow-based graph is crucial because a precise graph will result in a precise system state representation, thus leading to the policy updating and optimizing in an explicit and accurate manner. However, cloth folding highly relates to occlusion, and our current visual corresponding method can not deal with fully occluded objects. Furthermore, secondly, training such dense corresponding descriptors is time-consuming. On the other hand, how to imitate a high-speed motion (i.e., the robot in pushing sub-process requires a relatively high-speed motion) remains an open question, and little research relates to this topic at present. To fully solve this problem, multi-perspective or active vision might help collect more valuable observations from a vantage viewpoint, thus accelerating the policy searching process. Another interesting topic for future work would be encoding prior knowledge such as physical models into imitation learning policy learning.

## V. Conclusion

In this paper, we proposed a solution for cloth folding by learning from video demonstration using a refined optical flow-based STG as the input. With a dense corresponding descriptor, we identify the intended pixel between different video frames to construct a basic STG. Subsequently, we combine optical flow and static motion saliency map, aiming at refined STG with attentional motion mechanism. Experimental results on a real robotic platform have validated the effectiveness and robustness of the proposed approach. As future research, we plan to study how to optimize and update the policy learning from cloth folding with the assistance of a folding board to the task without the folding board.

## References


[1] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Sci. Robot.*, vol. 6, no. 54, 2021.

[2] S. Miller, J. Van Den Berg, *et al.*, "A geometric approach to robotic laundry folding," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 249–267, 2012.

[3] P. Zhou, J. Zhu, S. Huo, and D. Navarro-Alarcon, "Lasesom: A latent and semantic representation framework for soft object manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5381–5388, 2021.

[4] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 429–441, 2016.

[5] D. Navarro-Alarcon, Y.-h. Liu, *et al.*, "On the visual deformation servoing of compliant objects: Uncalibrated control methods and experiments," *Int. J. Robot. Res.*, vol. 33, no. 11, pp. 1462–1480, 2014.

[6] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *IEEE Trans. Robot.*, vol. 34, no. 1, pp. 272–279, 2018.

[7] Y. Li, Y. Wang, M. Case, S.-F. Chang, and P. K. Allen, "Real-time pose estimation of deformable objects using a volumetric approach," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*. IEEE, 2014, pp. 1046–1052.

[8] Y. Li, C.-F. Chen, and P. K. Allen, "Recognition of deformable object category and pose," in *ICRA*. IEEE, 2014, pp. 5558–5564.

[9] J. Borràs, G. Alenyà, and C. Torras, "Encoding cloth manipulations using a graph of states and transitions," *arXiv preprint arXiv:2009.14681*, 2020.

[10] B. Jia, Z. Pan, Z. Hu, J. Pan, and D. Manocha, "Cloth manipulation using random-forest-based imitation learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2086–2093, 2019.

[11] S. Donaire, J. Borras, G. Alenya, and C. Torras, "A versatile gripper for cloth manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6520–6527, 2020.

[12] I. Garcia-Camacho, M. Lippi, *et al.*, "Benchmarking bimanual cloth manipulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1111–1118, 2020.

[13] Y. Kuniyoshi, "Learning from examples: Imitation learning and emerging cognition," *Humanoid Robotics and Neuroscience: Science, Engineering and Society*, pp. 234–249, 2015.

[14] D. Pathak, P. Mahmoudieh, *et al.*, "Zero-shot visual imitation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 2050–2053.

[15] B. C. Stadie, P. Abbeel, and I. Sutskever, "Third-person imitation learning," *arXiv preprint arXiv:1703.01703*, 2017.

[16] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *CoRL*. PMLR, 2020, pp. 979–989.

[17] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[18] T. Osa, J. Pajarinen, *et al.*, "An algorithmic perspective on imitation learning," *arXiv preprint arXiv:1811.06711*, 2018.

[19] A. Hussein, M. M. Gaber, *et al.*, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[20] T. Zhang, Z. McCarthy, *et al.*, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 5628–5635.

[21] C. L. Nehaniv, K. Dautenhahn, and K. Dautenhahn, *Imitation in animals and artifacts*. MIT press, 2002.

[22] P. Sharma, D. Pathak, and A. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *NeurIPS*, 2019.

[23] M. Ollis, W. H. Huang, and M. Happold, "A bayesian approach to imitation learning for robot navigation," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*. IEEE, 2007, pp. 709–714.

[24] M. J. Zeestraten, I. Havoutis, J. Silvério, S. Calinon, and D. G. Caldwell, "An approach for imitation learning on riemannian manifolds," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1240–1247, 2017.

[25] D. Dwibedi, J. Tompson, *et al.*, "Learning actionable representations from visual observations," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*, 2018, pp. 1577–1584.

[26] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," in *ICML*. PMLR, 2018, pp. 4470–4479.

[27] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *ICCV*, 2017, pp. 2497–2506.

[28] A. B. Sargano, P. Angelov, and Z. Habib, "Human action recognition from multiple views based on view-invariant feature descriptor using support vector machines," *Applied Sciences*, vol. 6, no. 10, p. 309, 2016.

[29] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8, no. 2, pp. 179–187, 1962.

[30] E. Ilg, N. Mayer, *et al.*, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, 2017, pp. 2462–2470.

[31] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *CVPR*. Ieee, 2007, pp. 1–8.

[32] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 6, pp. 679–698, 1986.

[33] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine, "Combining model-based and model-free updates for trajectory-centric reinforcement learning," in *ICML*. PMLR, 2017, pp. 703–711.

[34] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 1134–1141.