This is the Pre-Published Version.

The following publication P. Zhou, J. Qi, A. Duan, S. Huo, Z. Wu and D. Navarro-Alarcon, "Imitating Tool-Based Garment Folding From a Single Visual Observation Using Hand-Object Graph Dynamics," in IEEE Transactions on Industrial Informatics, vol. 20, no. 4, pp. 6245-6256, April 2024 is avaliable at https://doi.org/10.1109/TII.2023.3342895.

Imitating Tool-based Garment Folding from a Single Visual Observation Using Hand-Object Graph Dynamics

***Abstract*—Garment folding is a ubiquitous domestic task that is difficult to automate due to the highly deformable nature of fabrics. In this paper, we propose a novel method of learning from demonstrations that enables robots to autonomously manipulate an assistive tool to fold garments. In contrast to traditional methods (that rely on low-level pixel features), our proposed solution uses a dense visual descriptor to encode the demonstration into a high-level *hand-object graph* (HoG) that allows to efficiently represent the interactions between the manipulated tool and robots. With that, we leverage Graph Neural Network (GNN) to autonomously learn the forward dynamics model from HoGs; then, given only a single demonstration, the imitation policy is optimized with a Model Predictive Controller (MPC) to accomplish the folding task. To validate the proposed approach, we conducted a detailed experimental study on a robotic platform instrumented with vision sensors and a custom-made end-effector that interacts with the folding board.**

***Index Terms*—Robotic manipulation, imitation learning, graph dynamics model, cloth folding, hand-object graph.**

I. Introduction

ROBOTS have been extensively used to support people in a variety of activities of daily living (ADL). Garment folding is a clear example of a monotonous service task that can theoretically be performed by robots but which, in practice, is difficult to solve by using these state-of-the-art strategies [1], [2]. One possible solution to alleviate the complexity of manipulating fabrics is to enable the robot to learn how to use an assistive tool by observing an expert demonstration and then imitating the behavior. This approach is typically referred to as imitation learning (IL) [3], [4], a technique that enables autonomous agents (e.g., robots) to acquire complex skills from simple sensory data without requiring to hard-code the strategies. Our aim in this work is to solve the garment folding problem by using an assistive tool under the imitation learning paradigm.

In contrast with other tasks solved by IL [5]–[7], the folding of garments is particularly challenging due to the high deformability and state dimension of fabrics. Previous works rely either on computationally expensive representation algorithms (i.e., based on polygonal models [8], particle-based models [9]) or elaborately designed manipulation pipelines for ad-hoc cases of study [10]. Furthermore, traditional IL approaches [11] assume that the state is fully observed (which is very difficult to satisfy in practice) and utilize model-free learning approaches (which typically demand large amounts of sample data). As classical imitation policies directly generate a mapping from the high-dimensional image space to the action space, they lack physical interpretability.

Our work is related to both imitation learning and model-based learning (see [11], [12] for comprehensive reviews). Imitation learning aims to learn a policy that can replicate expert behavior and generalize to unobserved states from several expert demonstrations comprised of state-action transitions. This approach has been demonstrated in many robotic tasks that are otherwise difficult to characterize, e.g., performing natural human-like arm movements [13], playing table tennis [5], and controlling the flight of a helicopter [6], to name a few cases. Although these methods can achieve satisfactory performance, they normally require demonstrations via tele-operation or kinesthetic teaching to generate state/observation-action transitions, which may not be possible to do in many applications [14]. Learning from expert video recordings (e.g., of a human operator folding garments) is a feasible alternative to this issue, however, this idea has not been sufficiently studied in the context of soft object manipulation tasks [15], [16].

Our aim in this work is to use imitation from observations [17] for tool-based garment folding, with a policy that must be learned from observations of a demonstrator's state transitions only. Therefore, model-based learning techniques [12] are needed to learn the underlying dynamics model by collecting action information. Though inverse reinforcement learning (IRL) [18] can solve imitation learning from observations, it typically learns the policy from a high-dimensional observation space (e.g., images), which has proven to be difficult to apply in practice and lacks interpretability of the learned policy. To deal with this, we seek to encode the dynamics model equation into a low-dimensional representation to help learn imitation policies with high interpretability. Compared with model-free approaches [19], model-based learning requires fewer interactions with the environment, thus, making policy learning more sample-efficient [20]; These learned models could be easily applied to new tasks [21]. Recently, several researchers have leveraged low-dimensional representations and dynamics models for policy learning, see e.g., [22], [23], but robot end-effector representations and dynamics are not considered which is critically important for imitation learning on robotic manipulation tasks.

To address the above-mentioned issues, in this paper, we formulate our proposed solution as a POMDP and employ a view-invariant dense descriptor model for detecting visual correspondences of the manipulated object (that is an assistive tool for garment folding); By introducing an assistive folding board, both manipulation efficiency and folding task quality are guaranteed. More importantly, given that the predefined points on the folding board can largely reduce state dimensions

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

and object deformability, this strategy enables the encoding of the state observation into a high-level (but low-dimensional) graph-structured representation, here called a *hand-object graph* (HoG) model. The vertices of the HoG denote the corresponding key points of the hand and the manipulated object (viz., a folding board), whereas the edges represent their relative 3D spatial configuration. We use a graph neural network to build a forward dynamics model over the HoG to predict transitions between HoGs and robot actions. During the one-shot imitation, for each time step, the learned dynamics model is used to optimize robot actions as the constraint for a model predictive controller that solves the task with a folding board in a closed-loop manner. While fixed automation (e.g., simple mechanical folders[1,2]) will continue to serve a role, our vision-based robotic system offers key advantages for handling the complexity, uncertainty, and frequent variations involved in garment manufacturing. By learning diverse skills from demonstration, these systems can achieve the re-configurability and adaptability that the industry requires. Experimental results on a robotic platform are carried out to demonstrate the effectiveness of our proposed framework. The original contributions of this work are summarized as follows:

- A new approach of introducing an assistive folding board to improve the performance of robotic garment folding tasks.
- A novel *hand-object graph* for high-level representation of robotic tool manipulation tasks.
- A new framework of imitation learning from a single observation based on graph-based forward dynamics model.

The rest of this article is organized as follows. Section II describes the problem definition. Section III presents the proposed framework for imitating garment folding tasks. Section IV shows the experimental results. Section V gives final conclusions.

For the convenience of readers, the main mathematical variables used in the article are given in Tab. I.

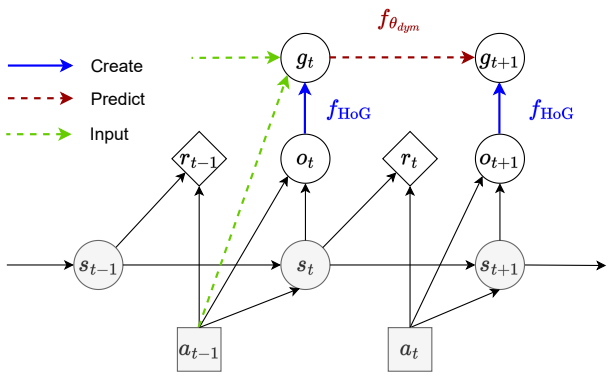

Fig. 1. **Reformulation.** The reformulated POMDP model after introducing the graph structure, which is generated with f_{HoG} and governed by $f_{\theta_{\text{dym}}}$.

II. Problem Formulation

We formulate the task of robotic garment folding as a Partially Observable Markov Decision Process (POMDP),

[1]https://www.youtube.com/shorts/X-I2AnT5ajM
[2]https://www.youtube.com/watch?v=OwwrxIShuU8

TABLE I
Mathematical Notations

Notation	Explanation
S	The state space.
A	The action space.
T	The state transition model.
O	The observation space.
B	The observation function.
r	The reward function.
γ	The discount factor.
$\pi(a_t\|o_t)$	The policy.
ξ_i	The i-th expert demonstration.
o_i^*	The i-th observation-only expert demonstration.
$\mathcal{D}$	The observation-only expert demonstration.
$\mathcal{G}$	The *hand-object graph* (HoG).
$f_{HoG}(\cdot)$	The HoG construction model.
$f_{\theta_{dym}}(\cdot)$	The forward dynamics model.
$\mathcal{O}^D(t)$	The t-step demonstration observation.
$\mathcal{O}^R(t)$	The t-step robot observation.
$\mathcal{R}(t)$	The t-step robot action.
$\mathcal{G}^R(t)$	The t-step HoG from robot observation.
$\mathcal{G}^D(t)$	The t-step HoG from demonstration observation.
H	The image height.
W	The image width.
C	The image channel.
$\mathcal{O}_{img}$	The image observation space.
$\mathcal{O}_{hand}$	The hand related image observation space.
$\mathcal{O}_{obj}$	The manipulated object-related image observation spaces.
$\mathcal{Z}$	The low-dimensional latent space.
$f_{\theta_{\text{dense}}}$	The dense descriptor function.
I_i	The i-th image.
M_i	The corresponding model in i-th image.
m	The pairwise pixel-level margin.
$\mathbf{x} = (x, y)$	The coordinate location in the image.
f_{C}	The correspondence function.
$\boldsymbol{d}_i$	The i-th dense descriptor.
$V(\mathcal{G})$	The vertex set of a graph.
$E(\mathcal{G})$	The edge set of a graph.
$\mathfrak{R}$	The linking rule between different sets.
$\mathcal{G}'$	The latent graph.
V'	The latent vertex set.
E'	The latent edge set.
φ^{v*}	The encoder function for latent vertices.
ϑ^{e*}	The decoder function for latent edges.
f_{θ_E}	The message passing function for latent edges.
f_{θ_V}	The message passing function for latent vertices.

which can be specified by a tuple $(S, A, T, O, B, r, \gamma)$, where the state $s_t \in S$ is unknown and can be denoted as the configuration of robots and manipulated objects, and its corresponding observation is $o_t \in O$. State transition model $T(s_{t+1} \mid s_t, a_t)$ characterizes the probability of a transition from state s_t to state s_{t+1} after executing action $a_t \in \mathcal{A}$, and observation function $B(o_t \mid s_t, a_{t-1})$ represents the probability that observation o_t will be recorded after performing action a_{t-1} and landing in state s_t. $r(s_t, a_t) \in \mathbb{R}$ is a state reward function, and $\gamma \in [0, 1)$ is the discount factor.

Imitation learning is considered to be defined in the context of a POMDP without an explicitly-defined reward function. It aims to seek a policy $\pi(a_t|o_t) : O \to A$, that the agent should take to behave like the expert demonstrations $\{\xi_1, \xi_2, \ldots\}$, where each ξ corresponds to a demonstrated observation action trajectory $\{(o_0, a_0), (o_1, a_1), \cdots, (o_n, a_n)\}$. In our setting, the imitator does not have access to the action sequences from the human demonstration and only one demonstration is available,

so the resulting imitation learning problem is referred to as *one-shot imitation learning from observation*. As such, our imitation learning goal is to obtain an imitation policy from a single observation-only demonstration $\{\xi\}$, where ξ is an observation-only trajectory $\{(o_i)\}$.

Given a single observation-only demonstration denoted as $\{o_1^*, o_2^*, \cdots, o_n^*\} \in \mathcal{D}$, we design a semantic structure comprising low-dimensional features extracted from the observation, which can visually describe the robotic manipulation process in order to enhance the interpretability of the learned policy. Specifically, the observation is partitioned into two sections, which are robot-specific features and object-specific features, respectively. Based on this insight, we introduce a *hand-object graph* (HoG), $\mathcal{G}$, which is composed of the robot hand feature and the manipulated object feature. As such, we reformulate the original POMDP as $(S, A, T, O, B, r, \gamma, \mathcal{G})$ (see Fig. 1), where $g_t \in \mathcal{G}$ can be constructed with a HoG model, $f_{HoG}(o_t)$ from the observation o_t, and governed by the learned forward dynamics model characterized by $f_{\theta_{dym}}(g_{history}, a_{t-1})$, where $g_{history}$ denotes a short history of g. Therefore, the goal of our task can be reformulated as $\pi : \mathcal{G} \rightarrow A$, where $g_t = f_{HoG}(o_t)$. In general, the specific problems that we are interested in are as follows:

- How to generate low-dimensional features from raw expert demonstration?
- How to form a valuable semantic structure to model the robotic manipulation process?
- How to train a dynamics model over high-level structured representations to accomplish our tasks?

III. Proposed Framework

As depicted in Fig. 2, we now present our framework of learning from observation for the garment folding task, which is mainly composed of three modules, namely, *hand-object graph* (HoG) construction, forward dynamics model, and one-shot learning. A dense 3D reconstruction of the folding board using an automatic style is designed to generate the training data set for visual correspondences first. Followed by a dense descriptor model, which is trained to capture the visual correspondence relationships between different images. Subsequently, combined with a pre-defined descriptor set, a *hand-object graph* is constructed. Then, a forward dynamics model is built with a graph neural network purely based on HoGs over 2D images. Finally, one-shot imitation is implemented using an MPC to optimize the robot action on 3D HoG with aligned RGBD images.

Given a single garment folding demonstration denoted as observations, $\{\mathcal{O}^D(1), \mathcal{O}^D(2), \cdots, \mathcal{O}^D(n)\} \in \mathcal{D}$, considering the current time step t, let the current observation for the demonstrator and the robot be $\mathcal{O}^D(t)$ and $\mathcal{O}^R(t)$, respectively. Our goal is to seek a robot action to reach $\mathcal{O}^R(t+1)$ by imitating the next demonstration observation $\mathcal{O}^D(t+1)$. With our proposed framework, we can transform $\mathcal{O}^R(t-\lambda:t)$ (a robot observation history from $t-\lambda$ to t) and $\mathcal{O}^D(t+1)$ into $\mathcal{G}^R(t-\lambda:t)$ and $\mathcal{G}^D(t+1)$ using *hand-object graph* construction module. Then, taking the forward dynamics model as the constraint of the MPC, the robot action $\mathcal{R}(t)$ will be optimized by computing the difference between resulting $\mathcal{G}^R(t+1)$ and $\mathcal{G}^D(t+1)$ in a closed loop manner.

A. Dense Descriptor Model

In practice, we extract the *hand-object graph* from the originally high-dimensional image observation space $\mathcal{O}_{img} = \mathbb{R}^{H\times W\times C}$ with a resolution of $H \times W$ and channel number of C. This $\mathcal{O}_{img}$ can be divided into robot hand related part $\mathcal{O}_{hand}$ and manipulated object related part $\mathcal{O}_{obj}$. The observation of the robot hand is measured by a self-designed end-effector as described in Fig. 7, while for manipulated object observation we need to use a visual model to process and reduce into a low-dimensional space $z \in \mathcal{Z}$. Specifically, we aim to learn an imitation policy that can directly manipulate objects in RGB images. Based on [24], we can learn a pixel-level dense descriptor of the manipulated object and efficiently detect the correspondences between RGB images.

To simplify, let $I = o_{obj} \in \mathcal{O}_{obj}$ be an image with a resolution of $H \times W$ and channel of C, and $f_{\theta_{\text{dense}}}$ dense descriptor without modifying its resolution. Our goal is to learn a dense descriptor with D dimensions $f_{\theta_{\text{dense}}}(I) : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times D}$ to assign a unique identifier to each pixel. In order to expedite the training procedure, we used the backbone of Fully Convolutional Networks [25] to reapply calculated activations for overlapping pixels, thereby effectively obtaining the dense descriptor. For the garment folding task, we collect an N-frame image observation sequence, $\{(I_1, M_1), \cdots, (I_N, M_N)\}$, in which M_i is a global correspondence model of the image I_i to maintain mappings from pixel coordinates in the current frame to the corresponding coordinates in other frames. By automatically collecting RGB-D images with a robot arm, we can build a dense 3D reconstruction of the working environment and then capture pixel coordinate mappings and track them throughout the images [24]. Fig. 3 visualizes a sample trajectory of a particular corresponding model through a folding task. Note that only the corresponding model of the folding board is considered since the garment will deform into numerous configurations, thus not being able to provide a stable mapping. Therefore, we employ pairwise labelling for correspondence pixel coordinates and a contrastive loss [25] at the pairwise pixel level, which is specified as follows:

$$\mathcal{L}(f_{\theta_{\text{dense}}}(I_{\mathbf{x}}), f_{\theta_{\text{dense}}}(I'_{\mathbf{x}'}), M(I_{\mathbf{x}}), M(I'_{\mathbf{x}'})) = \begin{cases} \|f_{\theta_{\text{dense}}}(I_{\mathbf{x}}) - f_{\theta_{\text{dense}}}(I'_{\mathbf{x}'})\|^2, \text{if } M(I_{\mathbf{x}}) = M(I'_{\mathbf{x}'}) \\ \max(0, m - \|f_{\theta_{\text{dense}}}(I_{\mathbf{x}}) - f_{\theta_{\text{dense}}}(I'_{\mathbf{x}'})\|)^2, \text{if not} \end{cases} \quad (1)$$

where $f_{\theta_{\text{dense}}}(I_{\mathbf{x}})$ and $f_{\theta_{\text{dense}}}(I'_{\mathbf{x}'})$ are the dense descriptors for image I located at $\mathbf{x} = (x, y)$ and image I' located at $\mathbf{x}' = (x', y')$, respectively. If coordinates $\mathbf{x}$ and $\mathbf{x}'$ correspond to the same 3D points located on the manipulated object, we consider them as a positive pair and reduce the distance in the feature space. If not, the contrastive loss will separate them by at least a m margin.

With the built dense descriptor model, a set of K fixed descriptors $\{\boldsymbol{d}_i\}_{i=1}^K$ is introduced to track a set of points located on the manipulated object in image space. For each $\boldsymbol{d}_i$, it represents a vector in the D-dimensional descriptor space

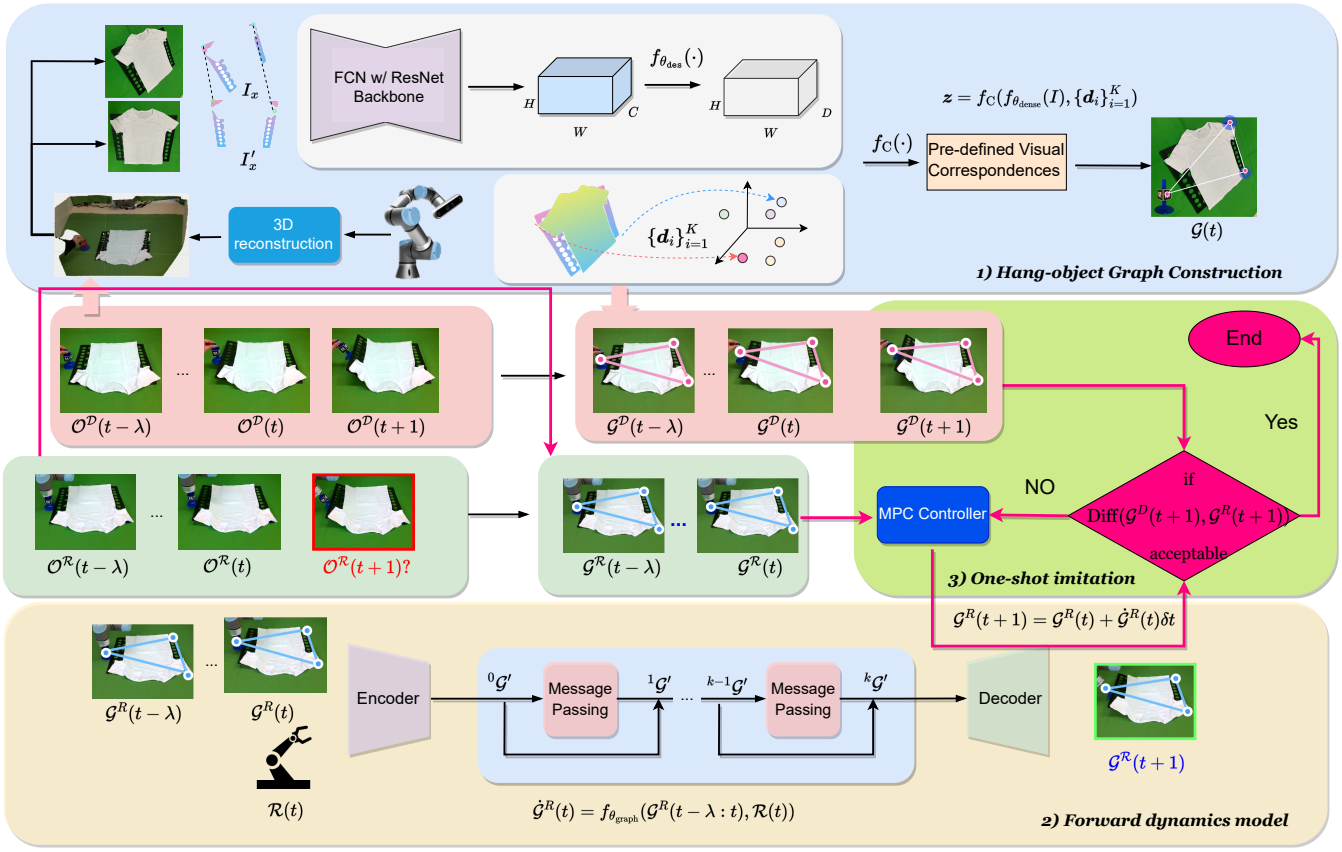

Fig. 2. **Overview.** The proposed framework of tool-based imitation learning for garment folding task is composed of three modules, namely, a visual dense descriptor-based *hand-object graph* model (marked in blue), a forward dynamics model built upon a graph neural network (marked in yellow), and a one-shot imitation learning module (marked in green). Given a human demonstration of garment folding task $\mathcal{O}^{\mathcal{D}}$ and current robot observation $\mathcal{O}^{\mathcal{R}}(t)$, the *hand-object graph* model constructs a high-level graph $\mathcal{G}^{\mathcal{D}}(t+1)$ based on $\mathcal{O}^{\mathcal{D}}(t+1)$, and graph $\mathcal{G}^{\mathcal{R}}(t)$ based on $\mathcal{O}^{\mathcal{R}}(t)$, followed by a forward dynamics model to generate $\mathcal{G}^{\mathcal{R}}(t+1)$. At last, the one-shot imitation will optimize the robot action until ending up with an acceptable graph difference between the human demonstration and predicted robot action denoted by $\text{Diff}(\mathcal{G}^{\mathcal{D}}(t+1), \mathcal{G}^{\mathcal{R}}(t+1))$.

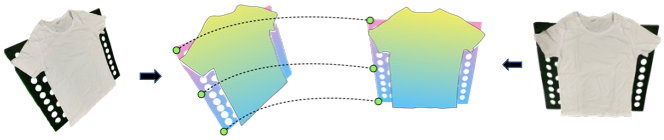

Fig. 3. **Mappings.** A visualization of the corresponding model from two different perspectives. The color map shows the pixel-level mapping provided by a dense 3D reconstruction of the working environment.

produced by the built dense descriptor model $f_{\theta_{\text{dense}}}(\cdot)$ that maps an RGB image $\mathbb{R}^{H\times W\times D}$ to a dense vector $\mathbb{R}^{K\times D}$. Let f_{C} denote the correspondence mapping function which takes dense vectors of the full-resolution image $f_{\theta_{\text{dense}}}(I)$ and the descriptor set as input to track the pre-defined points:

$$\boldsymbol{z} = f_{\text{C}}(f_{\theta_{\text{dense}}}(I), \{\boldsymbol{d}_i\}_{i=1}^{K}) \tag{2}$$

Note that the correspondence mapping function f_C has a consistent order of predefined points. In particular, $f_{\text{C}} : \mathbb{R}^{W\times H\times D} \times \mathbb{R}^{K\times D} \rightarrow \mathbb{R}^{K\times L}$, where $L = 2$ indicates the predicted pixel coordinates (x, y), while $L = 3$ represents their 3-dimensional coordinates.

B. Hand-object Graph Construction

Based on the tracked visual correspondence points, we propose a high-level hand-object graph (HoG) to characterize the interaction between the robotic end-effector and the folding board through the entire robotic manipulation process. Formally, a *hand-object graph* is a *partial join* graph $\mathcal{G} = \mathcal{G}_{hand} \vee \mathcal{G}_{obj}$ such that $V(\mathcal{G}) = V(\mathcal{G}_{hand}) \cup V(\mathcal{G}_{obj})$ and $E(\mathcal{G}) = E(\mathcal{G}_{hand}) \cup E(\mathcal{G}_{obj}) \cup \{uv \leftarrow \mathfrak{R} : u \in V(\mathcal{G}_{hand}), v \in V(\mathcal{G}_{obj})\}$, where $\mathcal{G}_{hand}$ is a hand graph generated by a convex polygon and $\mathcal{G}_{obj}$ a corresponding object graph, $V(\cdot)$ and $E(\cdot)$ denote the vertex set and edge set, respectively, and the connections between these two *disjoin* graphs are governed by the rule set $\mathfrak{R}$. In the interior of each graph $\text{Int}\,\mathcal{G}$, if there exist

Algorithm 1: *hand-object graph* Construction

Input: Observation at time step t, $\mathcal{O}(t)$
Output: *hand-object graph* at time step t , $\mathcal{G}(t)$.

for *each time step* t **do**
- $V(\mathcal{G}_{hand}) \leftarrow$ compute hand vertex set from $o(t)$;
- $V(\mathcal{G}_{object}) \leftarrow f_{\text{C}}(f_{\theta_{\text{dense}}}(\mathcal{O}_{img}(t)), \{\boldsymbol{d}_i\}_{i=1}^{K})$;
- $\mathcal{G}_{hand} \leftarrow \text{Convex\_hull}(V(\mathcal{G}_{hand})) \cup V_{isolated}$;
- $\mathcal{G}_{object} \leftarrow \text{Convex\_hull}(V(\mathcal{G}_{object})) \cup V_{isolated}$;
- $\mathcal{G}_{full} \leftarrow$ compute $\mathcal{G}_{hand} \vee \mathcal{G}_{obj}$ with rule set $\mathfrak{R}$;
- $\mathcal{G} \leftarrow$ refine $\mathcal{G}_{full}$;
- **return** *HoG* graph moment $\mathcal{G}(t)$;

isolated points, then they will be connected to their nearest neighbor until reaching the *convex hull* (please see [26] for details). For the hand graph $\mathcal{G}_{hand}$, we will select a point as the *master* vertex that remains relatively stationary to the robot flange, and others as *slave* vertices. The rule set $\mathfrak{R}$ is to connect the *master* vertex in the $\mathcal{G}_{hand}$ to each vertex in $\mathcal{G}_{obj}$ to construct their spatiotemporal relationships. Once the HoG is built, the relations between different vertices will be locked until the manipulation is completed. Based on HoGs, we define *Graph moment* as a spatio-temporal configuration of **HoG**, denoted as $\mathcal{G}(t)$ at time step t. As such, the difference between graph moments can be defined as the summation of the length differences of the corresponding edges comprising the HoGs:

$$\text{Diff}(\mathcal{G}(a), \mathcal{G}(b)) = \sum_{e_i \in E(\mathcal{G}(a)), e_j \in E(\mathcal{G}(b))} \|(e_i(a) - e_j(b))\|_2^2 \tag{3}$$

where e_i and e_j are the edges from the edge set of $\mathcal{G}(a)$ and $\mathcal{G}(b)$. By doing so, it only preserves the relative distance between the nodes instead of the specific coordinates of every node. The benefit is that the learned manipulation policy is able to work across different initial configurations and perspectives when taking this graph difference as its loss function. In practice, a full HoG $\mathcal{G}_{full}$ is not always necessary. For example, when we try to fold the left side, not all four corner points are needed. Thus, we often use a well-refined sub-graph $(\mathcal{G} \subseteq \mathcal{G}_{full} \mid E(\mathcal{G}) \subseteq E(\mathcal{G}_{full}), V(\mathcal{G}) \subseteq V(\mathcal{G}_{full}))$ that is still able to model the manipulation process correctly and we name it as *manipulation equivalence* of (**HoG**), which is very practical for real robotic manipulation tasks because a comprehensive observation of the manipulated object is not always feasible due to various occlusions in the working environment. The selection of hand vertex set and object vertex set, and HoG refinement are task-specific since they need to consider the manipulation context for different tasks, which is of significant importance, and we will put this in our future work.

As shown in Fig. 4, the selection of the hand vertex $V(\mathcal{G}_{hand})$ depends on the end-effector design. If a bimanual-manipulator is implemented to solve the task, then the number of the $V(\mathcal{G}_{hand})$ should be doubled. While the object vertex $V(\mathcal{G}_{object})$ could be a series of object landmarks and the selection depends on the inherent property of the manipulated object. If a rigid object is manipulated, then few points are enough, however, more dense points are necessary for soft objects. The advantage of this abstract visual structure is that it aligns with our human imitation learning mechanism, thus making the learned policy for the manipulation process much more explainable. Furthermore, this approach is independent of its workspace and will be much easier for policy transfer learning. In this work, as shown in Fig. 4 four key points representing the corner of the folding board and one end-effector point are considered to construct the hand-object manipulation graph. The corresponding **HoG** construction is illustrated in Alg. 1.

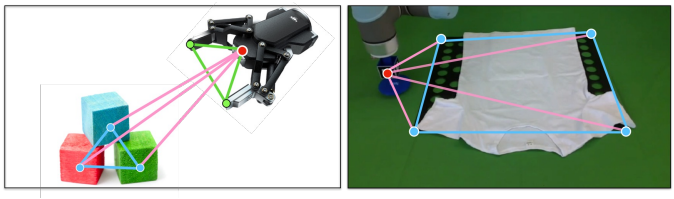

Fig. 4. **Graph Modeling.** Examples of *Hand-object graph* for manipulation tasks. (Blue convex polygon: hand graph $\mathcal{G}_{hand}$; green convex polygon: $\mathcal{G}_{obj}$; Red point: the *mater* vertex; Pink line: connections governed by rule set $\mathfrak{R}$).

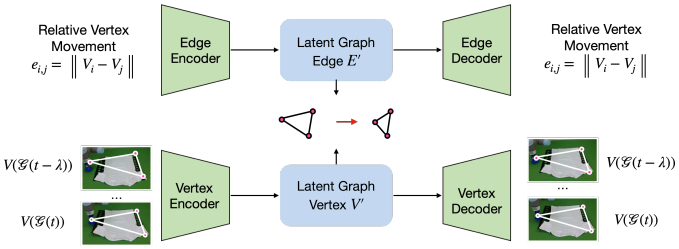

Fig. 5. **Latent Graph.** Conceptual representation of the encoder and decoder of *hand-object graph*, where the reconstruction loss is defined in Equ. (5) the original graph edges and vertices are transformed into a latent graph $\mathcal{G}'$ to simulate the forward dynamics, and then decoded back to the original space.

C. Forward Dynamics Model

Previous research [27] has shown the effectiveness of graph neural network (GNN) for complex dynamics learning. Similarly, we apply GNN to simulate the forward dynamics of the *hand-object graph* for the manipulation task. We define the dynamics model $f_{\theta_{\text{graph}}}(\cdot)$ to describe the interactions between the robot hand and the manipulated object as follows:

$$\dot{\mathcal{G}}(t) = f_{\theta_{\text{graph}}}(\mathcal{R}(t), \mathcal{G}(t - \lambda : t)) \tag{4}$$

where it takes the partial recent history of the *hand-object graph* $\mathcal{G}(t-\lambda : t)$ and robot velocity $\mathcal{R}(t)$. Without loss of generality, $\mathcal{R}(t)$ can denote multiple manipulators as $\{R_i(t)\}_{i=1}^{\Omega}$, in which each $R_i(t) \in \mathbb{R}^{1\times 6}$ represents the velocity of the end-effector for the i-th manipulator, and Ω is the total number of manipulators. We employ the GNS [28] network to characterize the HoG dynamics, which can be partitioned into three components - encoder, processor, and decoder: *a)* Encoder: Encoder is used to reduce the original high-dimensional space into a low-dimensional latent graph space $\mathcal{G}' = (V', E')$. As shown in Fig. 5, we train a graph edge encoder and graph vertex encoder, respectively, and the corresponding loss function is defined based on its reconstruction error as below:

$$\begin{aligned} \varphi^{e*}, \vartheta^{e*} &= \arg\min_{\varphi^e, \vartheta^e} \|E - (\varphi^e \circ \vartheta^e)(E)\|^2 \\ \varphi^{v*}, \vartheta^{v*} &= \arg\min_{\varphi^v, \vartheta^v} \|V - (\varphi^v \circ \vartheta^v)(V)\|^2 \end{aligned} \tag{5}$$

where $\varphi^{v*} : V \to \mathcal{Z}^V, \varphi^{e*} : E \to \mathcal{Z}^E$ are the encoders of the *HoG* edges and vertices, and $\vartheta^{v*} : \mathcal{Z}^{\mathcal{V}} \to \mathcal{V}, \vartheta^{e*} : \mathcal{Z}^{\mathcal{E}} \to \mathcal{E}$ are the corresponding decoders. *b)* Processor: we use the processor to compute the interactions between latent vertices and latent edges in order to simulate the HoGs' dynamics.

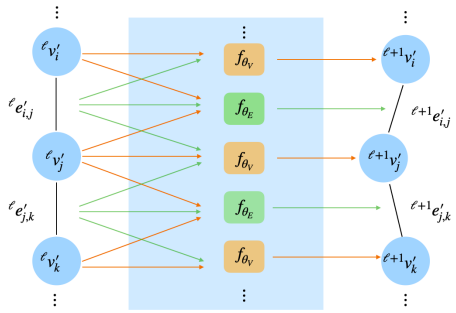

Fig. 6. **Message Passing.** Conceptual representation of the GNN message passing block. In each block, the interaction from one graph vertex to another is passed through the graph edge.

As shown in Fig. 6, the latent graph dynamics in the $(\ell+1)$-th message passing block can be computed as:

$$
\begin{aligned}
{}^{\ell+1}e'_{i,j} &= f_{\theta_E}\left({}^{\ell}v'_i, {}^{\ell}e'_{i,j}, {}^{\ell}v'_j\right) \\
{}^{\ell+1}v'_i &= f_{\theta_V}\left({}^{\ell}v'_i, \sum_j {}^{\ell}e'_{i,j}\right)
\end{aligned} \tag{6}
$$

where ${}^{\ell+1}e'_{i,j}$ and ${}^{\ell+1}v'_i$ denote the latent edge and vertex features in the $(\ell+1)$-th message processor, and f_{θ_E} and f_{θ_V} represents the latent graph edge network and vertex network, respectively. *c)* Decoder: As the last component, the decoders ϑ^V and ϑ^E are used to transform the latent graph edge and vertex features back to the original space to obtain the prediction $\dot{\mathcal{G}}(t)$, with a fixed weight after training.

Algorithm 2: One-shot Imitation Learning with *HoG*

Input: Graph dynamics model, $f_{\theta_{\text{graph}}}$; Expert Demonstration, $\{o_1^*, o_2^*, \cdots, o_n^*\} \in \xi$; Current observation, o_t; Robot velocity range, ϵ.

Output: A close-loop imitation policy π.

Initialize $\mathcal{R}$, ε ;

for *each time step* t **do**

  $\mathcal{G}(t) \leftarrow$ Compute imitation HoG based on o_t ;

  $\mathcal{G}(t+1)^* \leftarrow$ Compute corresponding demonstration HoG based on o_{t+1}^* ;

  $\mathcal{R}(t) \leftarrow$ Solve the optimization in Equ. (7);

  $o_{t+1} \leftarrow$ Execute $\mathcal{R}(t)$ and obtain the new observation ;

D. One-shot Imitation Learning

The use of a folding board to manipulate garments requires precise velocity control. Excessively high velocities can displace the entire garment, while overly low velocities lead to uneven folds. To enable accurate velocity control learned from human demonstrations, we employed model predictive control (MPC) with a closed-loop controller that optimizes end-effector velocity $\mathcal{R}(t)$ over a finite time horizon μ:

$$
\begin{aligned}
\min_{\mathcal{R}(1:\mu)} & \sum_{t=1}^{\mu+1} \text{Diff}(\mathcal{G}(t), \mathcal{G}(t)^*) \\
\text{s.t. } & \mathcal{G}(t+1) = \mathcal{G}(t) + \dot{\mathcal{G}}(t)\delta t \\
& \dot{\mathcal{G}}(t) = f_{\theta_{\text{graph}}}(\mathcal{R}(t), \mathcal{G}(t-\lambda:t)) \\
& \|\mathcal{R}(t)\| \leq \epsilon \\
& t = 1, 2, \ldots, \mu
\end{aligned} \tag{7}
$$

where the objective function is defined by minimizing the *HoG* difference defined in Equ. (3) and the constraints comprise the learned forward dynamics model and the robot's velocity $\|\mathcal{R}(t)\| \leq \varepsilon$, where δt is the model's sampling time. The entire procedure is presented in Alg. 2. During the imitating folding task, the optimal robot motions $\mathcal{R}(t)$ are iteratively solved by the MPC model and then transmitted to the robot for execution. The optimization is solved using a primal-dual interior-point linear search algorithm implemented by IPOPT [29]. As [30] has shown that local convergence is ensured, we primarily conducted experiments to empirically demonstrate the effectiveness.

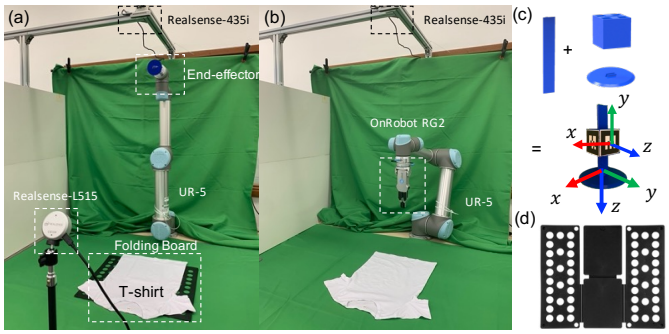

Fig. 7. **Experimental Setups.** (a) and (b) show experimental set-ups with a folding board and without a folding board. (c) and (d) present the custom-made end-effector for imitating the garment folding task with a standard folding board.

IV. Experiments

In this section, we outline the experimental setup for learning garment folding tasks, followed by implementation details of our visual dense descriptor model, hand-object graph modeling, and dynamics learning components. Finally, we compare our one-shot imitation approach to other imitation and fabric manipulation methods. As shown in Fig. 8, we aim to imitate three T-shirt folding tasks: **left folding**, **right folding**, and **middle folding**. For each task, we define four stages, namely, *approaching*: The end-effector moves toward the first contact point to insert the bottom circle board between the left side of the shirt and the workbench; *lifting*: The end-effector lifts to create enough space for rotating the garment; *rotating*: The end-effector rotates to align the folding board plane with the appropriate section of the garment; *pushing*: The end-effector pushes the corresponding section at a certain speed to complete the fold. To highlight the one-shot learning capabilities of our proposed **HoG** dynamics-based approach,

we provide one single expert demonstration video for each task. To examine the effectiveness and robustness of our approach, we also apply the imitation policy to the problem of different initial configurations, such as different rotations, garments, and perspectives.

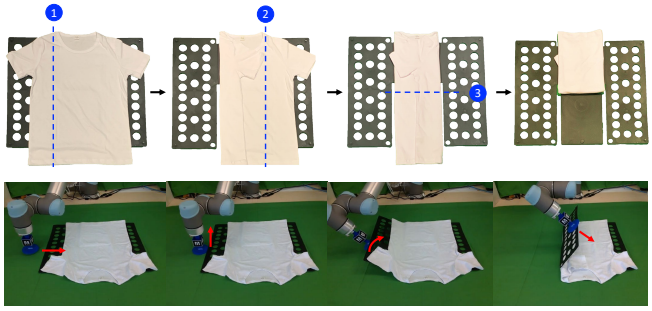

Fig. 8. **Folding Steps.** The first row shows three garment folding tasks with a folding board, which is composed of *left folding*, *right folding* and *mid-folding*. The second row displays the four procedures for each folding process, namely, *approaching*, *lifting*, *rotating* and *pushing*.

Fig. 9. **Initial Configurations.** (a) presents the four different initial configurations: (i) 0° *rotation angle*; (ii) -30° *rotation angle*; (iii) $+30^\circ$ *rotation angle*; (iv) *View-changed working space* for tool-based approaches (TCN, NoHoG and our full approach) in T-shirt and shorts folding task. (b) shows the configurations for non-tool-based imitation approaches (GSP and FFN).

A. Experiment Setup

Fig. 7(a) shows the experiment setup for our approach, where a RealSense RGB-D camera (L-515) is used to observe the garment folding process, and a UR-5 robot interacts with a black folding board with a custom-made end-effector, while 7(b) shows the experiment setup with selected solutions for comparing with our approach, where the folding board is removed and instead a robot gripper is used to pick and place garments. Both setups employ a RealSense RGB-D camera D435i to measure folding performance according to predefined metrics. We assume that the garment is initially laid out smoothly and flatly in the same place in the folder for each imitation episode. To precisely estimate the end-effector pose in our experiments, a modularized end-effector is designed (see Fig. 7(c)) which consists of a cube labelled with AR markers to detect its pose, a circular flake to interact with the black folding board (see Fig. 7(d)), and a stick to link them all. We calibrate the transformations from AR markers to the bottom circular plate's center. With such transformations, it is feasible to determine the position and orientation of the lower circular plate across the demonstrator's and imitator's workspaces, even when the assistive folder conceals the end-effector during the manipulation process.

B. Dense Descriptor Model

To construct the visual correspondence model, we need to collect images from different viewpoints based on a dense 3D reconstruction of the folding board. We place the folding board on the table at the front of the manipulator and scan it with an RGBD camera (RealSense D435i) mounted in a hand-eye style at 30Hz FPS from different viewpoints. This process takes about 60 seconds and saves approximately 1800 RGBD images. Based on the camera extrinsic calibration and forward kinematics of the robot, we acquire a global transformation of the captured RGBD images with respect to the robot base to obtain a dense 3D reconstruction. Next, we obtain new depth images by rendering the 3D folding board reconstruction back to each recorded camera frame using the estimated camera poses, which produces high-quality and globally consistent depth images to keep the pixel-level correspondence across RGB images for dense descriptor training. Since we only focus on the folding board to construct the following HoG, so the original RGB image will be converted into a mask RGB image by color filtering to accelerate the training process. Finally, around 7500 RGB images in the size of 640×480 are saved for dense descriptor training.

To train the dense descriptor model, a pretrained (on ImageNet) 34-layer, stride-8 ResNet is adopted, and then we bilinearly upsample to obtain an image with the original resolution. Networks are trained on a single Nvidia 1080 Ti using the same optimizer parameters ($Adam$ optimizer with a $1e-4$ weight decay) for 4,000 steps. In detail, each step requires approximately 0.34 seconds, i.e. approximately 22 minutes and the learning rate is set to $1e-5$ and drops by 0.8 per 250 iterations. Fig. 10(a) shows the dense descriptor for the four corner points of the folding board, where the first row presents the prediction results (red points and white lines) and ground truth (green points and blue lines), and the second row visualizes the corresponding heatmap of the dense descriptor. In addition, we record another video to measure its error during the manipulation process and Fig. 10(c) shows the corresponding pixel coordinate error, which is defined as $\|\mathbf{x}^{pred} - \mathbf{x}^{true}\|_2$. The most pixel coordinate error is below 2 pixels through the entire manipulation process, which is acceptable for our experiment considering that we always use the stationary corner points of the folder and the maximum 3D error is within 4 mm.

C. Graph Dynamics Model

The state of HoG is approximated by the visual correspondence model and the detected end-effector. We selected two corner points on the other side of the folding board. To interact with the folding board safely and validly, we design a rectangular bounding box in the 3D working space to ensure that the end-effector will not collide with the table surface. In addition, a predefined trajectory using explicit geometrical rule is executed when the end-effector is near the folding

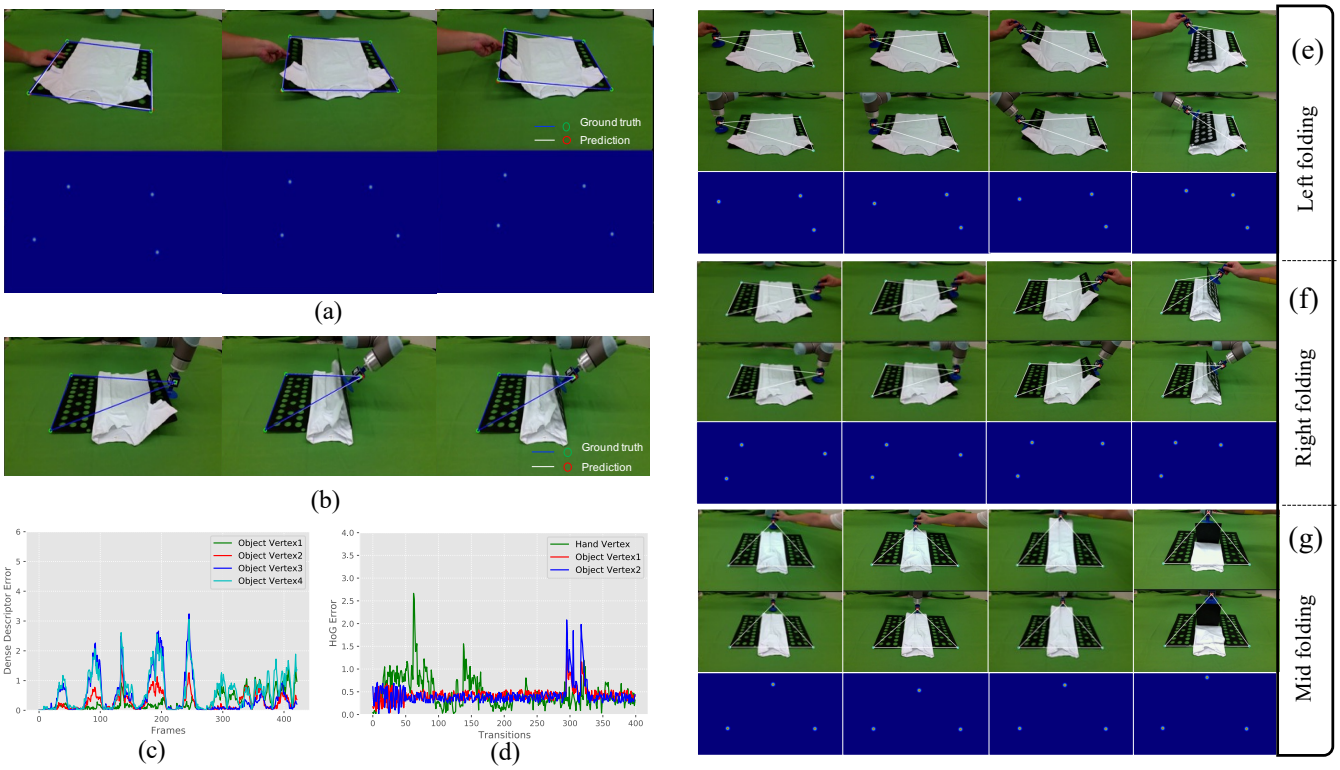

Fig. 10. **Results.** Performance of the the dense descriptor model, the forward dynamics model, and the process of imitating from observation of garment folding task. (a) and (b) present the ground truth and prediction of the dense descriptor model and the forward dynamics model; (c) and (d) show the corresponding error of the two models. (e)-(g) show successful imitation episodes for left-folding, right folding and mid-folding tasks, respectively.

TABLE II
PARAMETERS OF THE HOG FORWARD DYNAMICS MODEL

Dynamics Model Encoder Parameters	
Number of Hidden Layers	2
Size of Hidden Layers	64
Activation	ReLU
Dynamics Model Decoder Parameters	
Number of Hidden Layers	2
Size of Hidden Layers	64
Activation	ReLU
Dynamics Model Processor Parameters	
Number of Hidden Layers	2
Size of Hidden Layers	64
Activation	ReLU
Number of Message Passing (k)	10
Dynamics Model Training Parameters	
Batch Size	1
Learning Rate	10^{-4} to 10^{-6}
MPC Parameters	
Horizon of MPC (μ)	5
Sampling time of Dynamics (δt)	$0.1s$
optimizer	IPOPT

board to enable a successful inserting below the folder, so that the following interactions are valid. The folding board is initialized with a default configuration, and for each folding task we randomly execute the velocity commands for the end-effector (the maximum is 0.2m/s for x, y, and z-axis) to obtain 50 HoG-action trajectories $\{\mathcal{G}(t), \mathcal{R}(t)\}$, in which each trajectory has 100 transition steps and for each transition the 3D HoG configuration and the robot end-effector velocity are saved. By doing so, we form a data-set with 15,000 transitions. The network structure and MPC parameters are summarized in Table II. Similarly, we execute a predefined and unseen right folding trajectory to collect 400 HoG-velocity transitions to test the trained GNN dynamics model, and 10(b) and (d) show the performance and corresponding error, respectively. Finally, with the learned graph dynamics model, Fig. 10 (e)-(g) show the successful imitation episodes for left-folding, right-folding and mid-folding tasks, respectively. For model predictive control, we set the sampling time at 0.1 seconds. The total MPC duration depends on hand-object graph construction time (primarily from visual correspondence recognition), computation time for differences between the current and predicted graph states, optimization solving time using the learned forward dynamics model, and execution time for commands sent to the UR robot. We selected an MPC horizon length of 5 time steps based primarily on satisfying real-time constraints, given the complexity of optimization using our learning-based model at each control step. Longer horizons risked delays that could interfere with responsive control of the system. A horizon of H=5 was found to balance cost minimization over a reasonably significant prediction period with sufficiently fast performance for our application.

D. Comparison and Discussion

We compare our approach to three previous state-of-the-art techniques: (i) goal-conditioned skill policy (GSP) [32],

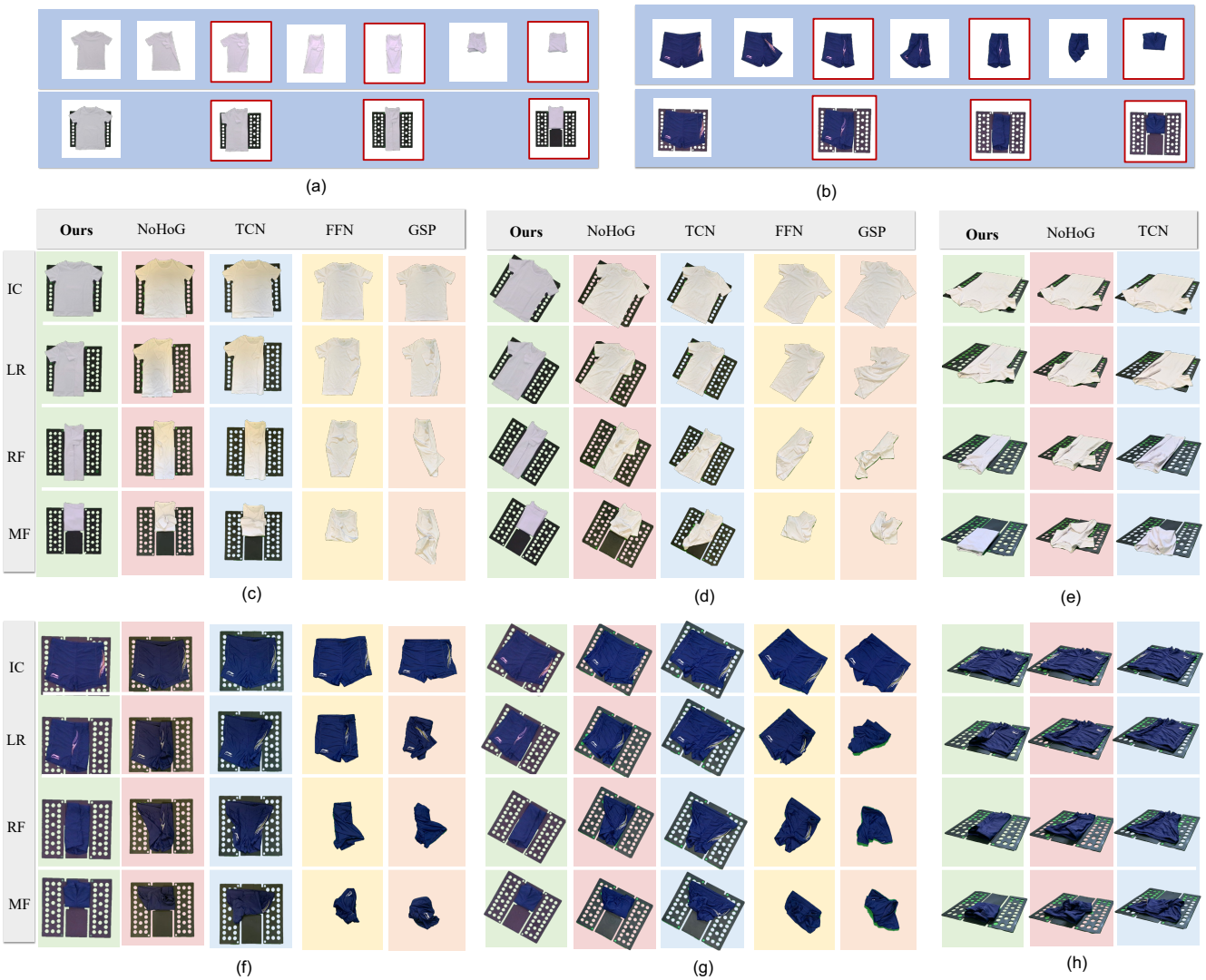

Fig. 11. **Qualitative Results in T-Shirt and Shorts Folding Experiments.** (a) and (b) show the sub-goal inputs for tool-based and non-tool-based baselines compared with our proposed approach, and the red rectangles represent the sub-goals where we will compute the task performance metrics (MIoU and Chamfer distance error). Three baseline approaches (GSP, FFN and TCN) and one ablation (NoHoG) are compared with our full approach in four different initial configurations for T-shirt folding tasks: (c) 0° *rotation angle*; (d) $\pm 30^\circ$ *rotation angle*; (e) *View-changed folding*. To evaluate the generalization ability, we also perform the folding tasks on unseen shorts with different approaches: (f) 0° *rotation angle*; (g) $\pm 30^\circ$ *rotation angle*; (h) *View-changed folding*. Note that since the performances of -30° *rotation angle* and $+30^\circ$ *rotation angle* are similar, only $+30^\circ$ *rotation angle* is reported in this figure.

TABLE III
MEAN IOU FOR FOLDING T-SHIRT AND SHORTS BY DIFFERENT METHODS

Method	T-shirt			Shorts		
	Rotation 0°	Rotation $\pm 30^\circ$	View Changed	Rotation 0°	Rotation $\pm 30^\circ$	View Changed
GSP [12]	0.42 ± 0.13	0.33 ± 0.21	/	0.14 ± 0.11	0.07 ± 0.04	/
FFN [11]	0.51 ± 0.24	0.43 ± 0.27	/	0.49 ± 0.28	0.35 ± 0.27	/
TCN [31]	0.39 ± 0.24	0.36 ± 0.23	0.42 ± 0.26	0.29 ± 0.24	0.31 ± 0.21	0.26 ± 0.25
NoHoG	0.49 ± 0.12	0.52 ± 0.16	0.41 ± 0.11	0.44 ± 0.19	0.37 ± 0.21	0.43 ± 0.22
HoG (Ours)	0.88 ± 0.03	0.83 ± 0.05	0.81 ± 0.07	0.75 ± 0.10	0.73 ± 0.09	0.72 ± 0.13

TABLE IV
MEAN CHAMFER DISTANCE ERROR FOR FOLDING T-SHIRT AND SHORTS BY DIFFERENT METHODS

Method	T-shirt			Shorts		
	Rotation 0°	Rotation ±30°	View Changed	Rotation 0°	Rotation ±30°	View Changed
GSP [12]	23.79 ± 12.74	24.71 ± 14.30	/	65.48 ± 31.81	72.54 ± 37.45	/
FFN [11]	16.87 ± 9.64	19.66 ± 10.67	/	23.10 ± 19.53	25.25 ± 21.63	/
TCN [31]	21.09 ± 9.22	24.89 ± 15.67	15.68 ± 6.49	28.12 ± 18.21	33.75 ± 19.20	38.66 ± 24.80
NoHoG	15.54 ± 7.32	18.77 ± 8.70	19.24 ± 9.26	19.18 ± 15.01	28.12 ± 16.63	34.07 ± 20.95
HoG (Ours)	11.64 ± 3.54	9.68 ± 4.90	10.60 ± 4.48	16.47 ± 7.30	15.55 ± 8.92	16.04 ± 7.75

TABLE V
SUCCESS RATE FOR FOLDING T-SHIRT AND SHORTS BY DIFFERENT METHODS

Method	T-shirt				Shorts			
	Rotation 0°	Rotation ±30°	View Changed	*Total*	Rotation 0°	Rotation ±30°	View Changed	*Total*
GSP [12]	9/30	23/60	/	35.56%	0/30	0/60	/	0.00%
FFN [11]	14/30	33/60	/	52.22%	12/30	26/60	/	42.23%
TCN [31]	19/30	32/60	17/30	56.67%	7/30	6/60	8/30	17.50%
NoHoG	21/30	43/60	10/30	61.67%	19/30	31/60	7/30	47.50%
HoG (Ours)	27/30	56/60	26/30	**90.83%**	23/30	52/60	22/30	**80.83%**

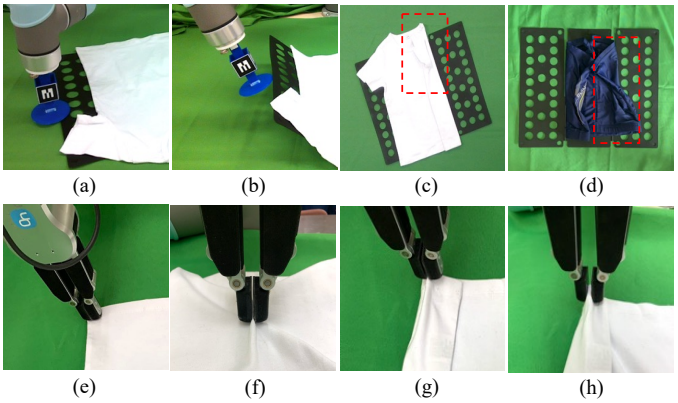

Fig. 12. **Failure Case.** (a)-(d) show the failure cases of error insertion, error rotation, small velocity and small velocity in our approach. (e)-(h) present the failure cases of unsuccessful picking, error picking position, one-layer picking, and multi-layer picking in FFN.

which learns a manipulation policy for imitation learning tasks; (ii) Fabricflownet (FFN) [33], which learns a goal-conditioned flow-based policy for fabric manipulations; (iii) time-contrastive networks (TCN) [31], which is a robotic imitation learning approach with a self-supervised paradigm. To further evaluate the importance of the component of our HoGs, we conduct an ablation analysis, which is a variant of our approach ("NoHoG") by replacing the HoG dynamics model with a naive forward dynamics model designed in GSP. As shown in Fig. 7(b), a non-tool-based experimental setup is used to evaluate the GSP and FFN (due to the pick-and-place action space) on folding tasks, which can be achieved with six actions. While the TCN, NoHoG, and our full approach are evaluated in a tool-based experimental setup (see Fig. 7(a)) for folding tasks, which can be achieved in three folding steps (see Fig. 11(a) and (b) for details).

Similar to previous work [33], we evaluate the folding task performance quantitatively by using Mean Intersection over Union (MIoU) between the garment masks achieved by different approaches and a human demonstrator. Since the MIoU metric only considers the intersection between two masked garment boundaries, to measure the surface flatness of folded garments achieved by different approaches and a human demonstrator, we select the Chamfer distance error [34] as another performance metric to compare extracted point clouds of the garment surface captured by the top-down depth camera. By using the farthest point sampling algorithm proposed in PointNet++ [34], the point clouds can be downsampled into a fixed resolution, which indicates that the total number of points in the point clouds is the same. Consider two downsampled point clouds $\mathcal{P}_i$ and $\mathcal{P}_j$ with N points in each cloud, the total Chamfer distance error is defined as:

$$\mathcal{D}\left(\mathcal{P}_i, \mathcal{P}_j\right) = \sum_{x \in \mathcal{P}_i} \min_{y \in \mathcal{P}_j} \|x - y\|_2^2 + \sum_{y \in \mathcal{P}_j} \min_{x \in \mathcal{P}_i} \|x - y\|_2^2 \quad (8)$$

where $x \in \mathcal{P}_i, y \in \mathcal{P}_j$. In our experiments, we set $N = 10000$ and the mean Chamfer distance error as $\mathcal{D}\left(\mathcal{P}_i, \mathcal{P}_j\right)/N$. To validate the effectiveness and robustness of our approach, as shown in Fig. 9 each folding task is performed under four different initial configurations: (i) $0°$ *rotation angle*; (ii) $-30°$ *rotation angle*; (iii) $+30°$ *rotation angle*; (iv) *View-changed working space*. Note that the GSP and FFN are not considered in *view-changed* configuration since they are trained under a fixed top-down view. For each configuration, a policy needs to be performed on two different garments (T-shirt and shorts) 30 times. For each policy, during training, we only provide T-shirt expert demonstration data. An episode is considered successful folding only when the MIoU ≥ 0.5 and the Chamfer distance ≤ 15.00, which are empirically set based on a large number of tests. During experiments, we found that the folding performance in initial configurations of $-30°$ *rotation angle* and $+30°$ *rotation angle* are very similar, so we report them in a combined manner with a new item named *Rotation* $\pm 30°$ in the following quantitative results.

Experimental quantitative results can be found in Table III, IV and V. Our full approach achieves the highest MIoU and lowest mean distance error over all tasks and shows a larger improvement over the baselines. Fig. 11 (c)-(h) show representative qualitative results under different initial configurations using different approaches. As can be observed, even when the garment's size, material, pose, and camera view are different from those of the demonstration process, our HoG-based approach performs all folding tasks successfully, while the baselines fail to complete most of the tasks. The high success rate of our proposed approach is mainly attributed to two aspects. First, introducing the folding board in the garment folding tasks is able to effectively constrain garment deformation into a reduced configuration space, thus better handling the complex dynamics of garments compared with non-tool-based approaches (GSP and FFN). Second, our full approach employs view-invariant spatio-temporal features to construct the dynamics model, thereby having the ability to handle folding tasks in various initial configurations, especially when the view changes (see Fig. 11 (e) and (h)). In addition, our HoG-based approach can also generalize to the shorts, which is unseen during training and demonstrations. By capturing view-invariant spatiotemporal features on the folding board, the proposed approach can understand high-level dynamics information about a multi-step task. Thus, our approach is robust to garment pose, camera view, and even garment type variations in the low-level pixel space.

GSP and TCN perform poorly on the shorts compared with their performance on the T-shirt because they are both built upon feature representation extracted from pure RGB images, which lacks the ability to transfer learned policies onto unseen shorts. However, FFN performs better than GSP on both the T-shirt and shorts, which is because FFN only uses depth images for both training and implementing processes, which can transfer the learned policy. A certain portion of the failure cases of FFN involve grasping failures as it uses pick-and-place action primitives (see Fig. 12 (e)-(h) for details). For the ablation, the "NoHoG" ablation validates the significance of providing an HoG-based dynamics model for better capturing the system dynamics, especially for view-changed initial configurations. Failure cases in our experiments mostly come from end-effector insertions, rotations, and inappropriate velocities. Since these actions are very sensitive and only allow for small tolerances, it's difficult to determine the most desired pose in a real-time manner. Fig. 12 illustrates these failure cases. In future work, we will introduce tactile sensory data to improve the action's quality and reliability.

V. Conclusion

In this paper, we propose a novel framework of imitation learning from a single observation to solve tool-based robotic garment folding tasks by using a graph-based dynamics model. With a dense descriptor, we encode the demonstration observation into a high-level *hand-object graph* for characterizing the interactions between the assistive tool and robots. After that, we construct a forward dynamics model with graph neural networks aiming at simulating the complex dynamics of the HoGs. Finally, robotic garment folding is solved with a model predictive controller under the paradigm of imitation learning from a single observation. The effectiveness of the proposed framework has been validated by qualitative and quantitative experimental results conducted on a real robotic platform. Currently, we assume that the *hand-object graph* is fully observed throughout the entire manipulation process and can be obtained by empirical and manual selection. Still, further studies are needed to assess the impact of partially observed HoGs on the noisy output of correspondence detection models. Future research will focus on removing manual interference for initial configurations, improving the HoG modeling and extending the HoG dynamics model to other robotic manipulation tasks.

References

[1] J. Borràs, G. Alenyà, and C. Torras, "A grasping-centered analysis for cloth manipulation," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 924–936, 2020.

[2] R. Jangir, G. Alenya, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2020, pp. 4630–4636.

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

[4] B. D. Argall, S. Chernova *et al.*, "A survey of robot learning from demonstration," *Rob. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009.

[5] J. Kober and J. Peters, "Learning motor primitives for robotics," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2009, pp. 2112–2118.

[6] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.

[7] Y. Zhang, F. Qiu, T. Hong, Z. Wang, and F. Li, "Hybrid imitation learning for real-time service restoration in resilient distribution systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 2089–2099, 2021.

[8] J. Stria *et al.*, "Garment perception and its folding using a dual-arm robot," in *IEEE/RSJ Int. Conf. on Robots and Intelligent Systems*. IEEE, 2014, pp. 61–67.

[9] Y. C. Hou, K. S. Mohamed Sahari, L. Y. Weng, D. N. T. How, and H. Seki, "Particle-based perception of garment folding for robotic manipulation purposes," *International Journal of Advanced Robotic Systems*, vol. 14, no. 6, p. 1729881417738727, 2017.

[10] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1461–1478, 2016.

[11] 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.

[12] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," *arXiv preprint arXiv:2006.16712*, 2020.

[13] A. Billard and M. J. Matarić, "Learning human arm movements by imitation:: Evaluation of a biologically inspired connectionist architecture," *Rob. Auton. Syst.*, vol. 37, no. 2-3, pp. 145–160, 2001.

[14] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *IJCAI*, 2018.

[15] 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.

[16] D. Huang, B. Li, Y. Li, and C. Yang, "Cooperative manipulation of deformable objects by single-leader-dual-follower teleoperation," *IEEE Transactions on Industrial Electronics*, 2022.

[17] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 1118–1125.

[18] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artif. Intell.*, vol. 297, p. 103500, 2021.

[19] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," *Artif. Intell. Rev.*, vol. 55, no. 4, pp. 3213–3241, 2022.

[20] 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 *Int. Conf. Mach. Learn.* PMLR, 2017, pp. 703–711.

[21] P. Englert, A. Paraschos, J. Peters, and M. P. Deisenroth, "Model-based imitation learning by probabilistic trajectory matching," in *IEEE Int. Conf. on Robotics and Automation*. IEEE, 2013, pp. 1922–1927.

[22] C. Wang, Y. Zhang *et al.*, "Offline-online learning of deformation model for cable manipulation with graph neural networks," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5544–5551, 2022.

[23] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 492–499, 2019.

[24] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conf. Rob. Learn.* PMLR, 2018, pp. 373–385.

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

[26] I. M. Pelayo, *Geodesic convexity in graphs*. Springer, 2013.

[27] J. Zhou, G. Cui *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[28] A. Sanchez-Gonzalez, J. Godwin *et al.*, "Learning to simulate complex physics with graph networks," in *Int. Conf. Mach. Learn.* PMLR, 2020, pp. 8459–8468.

[29] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Motivation and global convergence," *SIAM Journal on Optimization*, vol. 16, no. 1, pp. 1–31, 2005.

[30] ——, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[31] 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.

[32] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, "Zero-shot visual imitation," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 2050–2053.

[33] T. Weng, S. M. Bajracharya, Y. Wang, K. Agrawal, and D. Held, "Fabricflownet: Bimanual cloth manipulation with a flow-based policy," in *Conference on Robot Learning*. PMLR, 2022, pp. 192–202.

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.