A Novel Robust Imitation Learning Framework for Complex Skills With Limited Demonstrations

Weiyong Wang, Chao Zeng, *Member, IEEE*, Hong Zhan, and Chenguang Yang, *Fellow, IEEE*

***Abstract*— Imitation learning allows us to directly encode manipulation skills based on human demonstrations, facilitating rapid transfer of skills without any expert knowledge. Autonomous dynamic systems (DS) offer reliable stability and time-independence though sacrificing part of accuracy, and are increasingly attractive as an encoding method. As unstructured environments become more challenging, skill trajectories become more complex, and various disturbances are encountered, existing state-of-the-art encoding methods struggle to adapt to these complex tasks. This paper introduces a novel robust DS-based framework for learning skills in complex tasks, which consists of trajectory regularization, adaptive segmentation, skill modeling, and skill organization based on new task requirements. It achieves a task-level generalization so that the operator only needs to focus on the semantic deconstruction of a task. Additionally, we propose an online modulation policy for the skill decision engine to address two types of disturbances: enhancing convergence speed for large-scale disturbances and improving fitting capability for small-scale disturbances while still keeping stability. To evaluate the effectiveness of the proposed framework, we conduct various comparison experiments in simulation and a real-world sugar-scooping task to assess the generalization performance and the ability of resistance to disturbances.**

***Note to Practitioners*—Imitation learning for complex tasks is crucial to the development of robot intelligence. However, achieving a balance between maintaining generalization accuracy and robustness remains a challenging problem that necessitates continuous exploration in the field of imitation learning. The purpose of this paper is to propose a robust imitation learning framework from human demonstration, which includes preprocessing, learning and generalization, that can be applied in industrial production or daily life. Considering appropriate trajectory segmentation and self-organization strategies can effectively improve the generalization accuracy by prior research, it is necessary to introduce them into our framework. Most importantly, we design novel disturbance-resistant online modulation strategies from both task-level and motion-level aspects. To validate the effectiveness of our approach, we conduct simulations and coffee scooping experiments. The results show that skills acquired through demonstration can reliably, accurately, and safely perform tasks even in uncertain environments. This paper is a systematic and pioneering attempt to implement.**

***Index Terms*— Imitation learning, motion skills library, dynamic system.**

I. Introduction

As WE enter the fourth industrial revolution, conventional manufacturing is being steadily automated to the point that many repetitive production skills are programmed into robots. Imitation learning, also known as learning from demonstration (LfD) [1], [2], [3], is an important way for human-robot skill transfer. It is motivated by the process of human teaching and relies on human behavioral representation and imitation learning (IL). This greatly reduces the need for non-professional programming abilities and simply requires people to demonstrate by kinesthetic teaching, teleoperation, or passive observation [4], [5], [6].

Inspired by biology, a movement primitive refers to a relatively simple motion perception sequence that has been successfully used for a long period [7], [8]. It describes a behavioral unit with a unique attractor that can further compose complex behaviors and adapt to disturbances. However, with the development of industrial productivity and artificial intelligence, the requirements for robot skill learning are gradually increasing, including facing complex unstructured environments and learning more complex and more constrained skills. Namely, robots need to engage in flexible interactions and adapt to changes in humans, tools, or manipulated objects. The limitations of classical single-primitive modeling solutions for complex tasks have become increasingly evident.

A common solution for learning a complex task is to model a hierarchical logical graph [9], which is often referred to as a multi-step task. It usually requires abstraction from different levels, including task-level planning and motion-level execution. When faced with a multi-step task, previous research enables the robot to complete the task by dividing the demonstration trajectory into different target points [10], [11] or primitive phases [12], [13]. These methods serialize the execution of subtasks and solve the problem to a certain extent, but they still cannot guarantee the successful reproduction of the task. Such difficulties include: (i) automatically decomposing the task into segments suitable for learning; (ii) extracting uncertainty and analyzing the reliability of the control signal; (iii) the learned model guarantees convergence and trajectory shape; (iv) task-level and motion-level robustness. A highly

Manuscript received 17 December 2023; revised 26 February 2024; accepted 18 May 2024. Date of publication 24 May 2024; date of current version 21 February 2025. This article was recommended for publication by Associate Editor W. G. Guo and Editor J. Li upon evaluation of the reviewers' comments. This work was supported in part by the UKRI Postdoctoral Fellowships Guarantee under Grant EP/Z00117X/1 and in part by the State Key Laboratory of Robotics and Systems (HIT) under Grant SKLRS-2024-KF-09. *(Weiyong Wang and Chao Zeng contributed equally to this work.) (Corresponding author: Chenguang Yang.)*

Weiyong Wang and Hong Zhan are with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: 202120117998@mail.scut.edu.cn; auhzhan@scut.edu.cn).

Chao Zeng and Chenguang Yang are with the Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: chaozeng@ieee.org; cyang@ieee.org).

Digital Object Identifier 10.1109/TASE.2024.3403833

autonomous and adaptive IL paradigm is difficult to be still fully achieved through standard machine learning techniques.

To address the above challenges, in this paper, we develop an imitation learning framework that not only emphasizes skill transfer efficiency and capability but also considers dealing with potential disturbances during generalization. As Fig. 1, the framework architecture is divided into three parts:

A. Preprocessing

This part consists of trajectory regularization and segmentation. The raw data obtained from human demonstration often comes with noise, redundant segments or dimensions. Appropriate regularization standardizes the raw data into a unified format and improves the performance of subsequent learning algorithms. Commonly used regularization steps include alignment [14], smoothing filtering [15], and dimension screening [16]. Trajectory segmentation is a key process for converting complex skills into multi-step skills. We employ an improved Hidden Markov Model (HMM) to model the implicit discrete steps of the trajectory with the emission function in the form of a Gaussian function, which can capture the spatial dependencies of a signal. It adaptively and automatically divides trajectories into segments (motion level) and reveals the switching logic and steps (task level). Then, a complex task is decomposed into a set of simpler subtasks, which can then be modeled more effectively.

B. Learning

The role of learning is to convert abstract subtask skills into various parameters of the mathematical model to facilitate our further reproduction and modulation. The primitives are typically defined in terms of their input-output behaviors and can be learned by either deterministic or statistical modeling. The poses of the end-effector or the joint angle of the manipulator are common for model inputs, while the output can be a control signal such as velocity, acceleration, or torque. It is worthwhile to note that we avoid using time-driven models for acquiring spatial and temporal robustness. Here, we employ the Linear Parameter Varying Dynamical System (LPV-DS) for modeling, with Quadratic Lyapunov function (P-QLF) optimization constraints to ensure global asymptotic stability. The skill primitives are further collected and stored as a skill library, which enables quick and flexible reuse of motion primitives in the generalization process and greatly reduces the time and effort needed to learn similar tasks.

C. Generalization

Skill generalization refers to the ability of the learned skill to be applied to new situations that are different from the demonstration. We construct a Behavior Tree (BT) from task-level switching logic and steps to guide transitions between different primitives. Based on the current state, the skill decision engine selects an appropriate primitive from the skill library to execute and ensure the smooth completion of a task. The reproduction does not require modulation under a configuration that is similar to the demonstration. But once the trajectory deviates, the control output will become unreliable or even dangerous, and needs to be corrected in time. Therefore, we innovatively modulate the control output to withstand large-scale and small-scale disturbances, improving generalization from task level and motion level respectively.

Before proceeding with the introduction, it's essential to clarify two concepts: (i) **Subtask space**. Regarding multi-step skills, we believe that each primitive should have its confident execution space, called subtask space. When a subtask skill is executed far from the corresponding subtask space, it could affect the successful execution of the task or even be dangerous. Although it does not need to be too explicitly bounded, it should at least contain the demonstration region where the sub-skill is located. (ii) **Two types of disturbances**. When a disturbance occurs, we categorize it into two types based on the manipulator's movement. If the manipulator deviates from the subtask space, it is called a large-scale disturbance; otherwise, it is called a small-scale disturbance. Importantly, this definition distinguishes disturbances not by their intrinsic amplitude, which do not intuitively affect the task. When large-scale disturbances occur, the manipulator's behavior becomes unreliable, necessitating a prioritized return to the subtask space. When small-scale disturbances occur, we pay more attention to the generalization accuracy.

The rest of this paper is organized as follows: Section II reviews related works on imitation learning and the shortcomings of existing methods. Section III introduces the adaptive skill segmentation and skill decision engine in our framework, as well as the autonomous DS modeling approach. In Section IV, a detailed modulation design for the skill decision engine is proposed to resist disturbances in the generalization process. In Section V, we design comparative tasks in simulation for different framework modules and quantitatively analyze the generalization accuracy of the online modulation strategy for disturbances. The performance of our proposed imitation learning framework is further verified in the experimental scene of adding sugar to coffee, showing efficient spatial and temporal robustness. Section VI discusses the experiment results and finally concludes this paper.

II. Related Works

As a quick and efficient way to transfer human skills, LfD has developed rapidly in the past two decades. Dynamical Movement Primitives (DMP) [17] is a typical time-driven skill modeling method, which identifies a single demonstration trajectory as a nonlinear spring-damper system with a globally unique attractor. Although the standard DMP has several limitations, such as its incapacity to simultaneously extract shared information from multiple demonstrations or encode high-dimensional inputs or outputs, its practicality still attracts widespread attention. For its flexible extensibility, researchers combine coupling terms with control theory including obstacle avoidance, dual-arm cooperation, and contact force modification [18], [19], [20], [21], [22], [23] to achieve more functionalities. To address the limitations of DMP, some other popular time-driven methods include Probability Movement Primitives (ProMP) [24], Kernel Movement Primitives (KMP) [25] and Gaussian Process Movement Primitive (GPMP) [26] have successively been proposed for LfD, enriching primitives

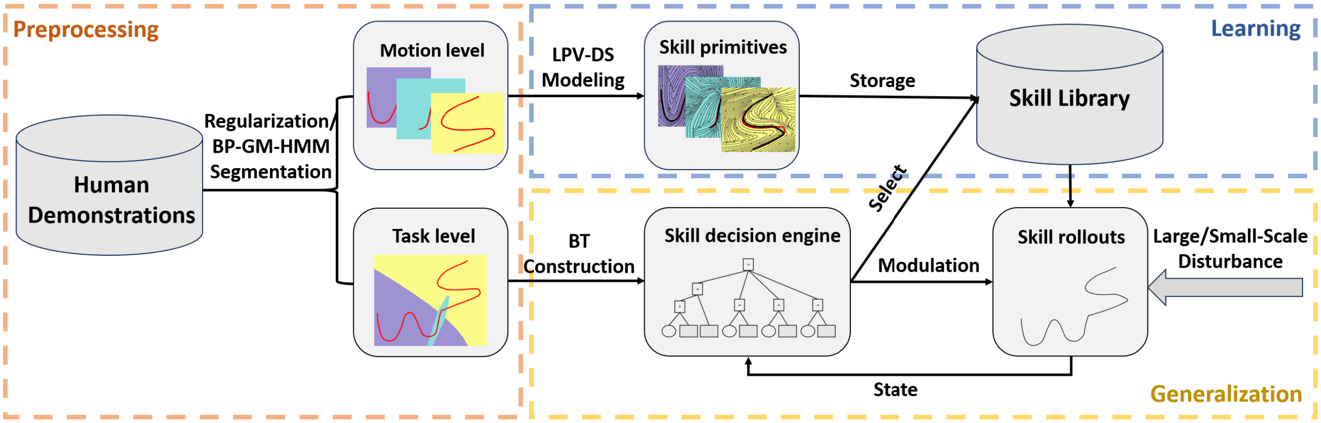

Fig. 1. The proposed framework for robot imitation learning includes three parts, i.e., preprocessing (Section III-A), learning (Section III-C) and generalization (Section III-B, IV). The preprocessing part regularizes the raw demonstration data, adaptively divides it into multiple trajectories (motion level) and reveals the switching logic between them (task level). The learning part converts the trajectories into various primitives and stores them as a skill library for quick and flexible reuse. The generalization part automatically constructs a BT according to the segmentation, then selects and applies primitives from the skill library to new situations. Importantly, it modulates the expression to resist large-scale and small-scale disturbances.

from high-dimension, variability, correlation. They all employ statistical noise to acquire the uncertainty of control signals to modulate primitives by the via points required for actual tasks or blending them. The above time-driven methods exhibit high generalization accuracies, however, due to their heavy reliance on the current time, they lack temporal robustness and are not able to flexibly adapt to unpredictable unstructured environments.

Regarding state-driven modeling, stable convergence and high accuracy has always been a challenging dilemma. Gaussian Mixture Model (GMM) [16] estimates the probability density distribution of the demonstration, and a conditional posterior distribution can be further obtained through Gaussian Mixture Regression (GMR). It can achieve excellent generalization accuracy by increasing the number of mixtures, yet its inability to stable convergence limits its application. Autonomous DS [27] is a state-driven differential equation, whose closed-form solution is determined by the initial state. Target convergence is typically studied by DS combined with contraction analysis [28] or Lyapunov functions [29] since it is essential for the safe and efficient execution of skills. Actually, contraction analysis has been proved to be a special condition of Lyapunov analysis [28], [30]. Although there is no general method to find Lyapunov candidate functions, the stability design based on Lyapunov functions has been widely used in the learning of DS due to its intuitive and effective characteristics. The Stable Estimator of Dynamical Systems (SEDS) [31] estimates nonlinear DS as a mixture of linear DS and parametrizes via GMM-GMR, rigorously guaranteeing global asymptotic stability through Lyapunov analysis. However, the overly strict selection of the Lyapunov function leads to poor performance in its generalization accuracy. Many efforts [32], [33] have been made to further relieve its accuracy limitations. [34] adopts the non-parametric Bayesian prior with physical consistent distance to improve the fitting ability of GMM. Compared with SEDS, it adaptively determines the mixture number of GMM, completely eliminating the need for human intervention. To improve the generalization accuracy of DS, it performs optimization with the Weighted Sum of Asymmetric Quadratic Function (WSAQF) [35], which is a less conservative Lyapunov function. Unfortunately, the generalization accuracy of a single primitive for complex nonlinear trajectories is still limited, which drives our work combining trajectory segmentation and primitive organization.

Appropriate decomposition for complex tasks can significantly enhance the generalization accuracy, reusability, and flexibility of motion-level skill primitives. Task decomposition and trajectory segmentation usually pay more attention to the extraction of segmentation points based on the distance to the task operation object [36] or the sensor state [37]. These methods rely on human intervention with prior knowledge and need to carefully design judgment conditions for each interactive object in a specific task. Regarding adaptive segmentation, [38] proposes a self-heuristic online segmentation method, which employs the prediction error of Linear Dynamical System (LDS) to identify a new segmentation. [39] represents a task as GMM, allowing incremental and online segmentation with GMR. However, the above segmentation methods cannot identify repeated subskills and are prone to be affected by hyperparameters. Reference [40] represents a task as Hidden Markov Model (HMM), and minimizes the loss error between the observation sequence and the demonstration trajectory to obtain the segmentation sequence. Since observations at different times can be manifested by the same latent state, repeated skills can be identified. References [41] and [42] further employ Beta Process Autoregressive Hidden Markov Model (BP-AR-HMM) to analyze common features of multiple sequences. The non-parametric Beta Process considers infinite clustering and thus does not require any prior knowledge anymore.

In fact, trajectory segmentation contributes to not only the motion-level generalization but also the task-level generalization. It often takes a lot of human effort to consider as many situations as possible for demonstration. But sometimes situations are complex or do not allow for this. The operators simply need to design specific switching logic to carry out task-level programming for various tasks. The Finite State Machines (FSM) [43] and BT [44] are the most common ways to build complex tasks, where tasks are described as state-charts flow or hierarchical tree structure respectively. Their powerful ability lies in the fact that program behavior is completely determined by internal logic states, allowing them to be combined to obtain robust behavior. The Linear Temporal Logic (LTL) formula [45], [46] is an abstract logical formula that can be effectively combined with automata-based specifications to provide a framework for rapidly constructing switching logic. This framework is advantageous for self-learning systems. Although the aforementioned methods differ in their descriptive form and task scalability, they are equivalent in the ability to describe the conversion logic. However, dealing with disturbances remains a significant challenge that has not been specifically addressed in research.

In this paper, we propose a novel robust imitation learning framework, which consists of the following key components to enhance the practicality for complex tasks: trajectory segmentation, skill decision engine, and stably convergent primitive modeling. Unlike common organizational methods based on task-level logic, the skill decision engine here not only selects primitives based on environmental perception but also modulates the expression of primitives according to disturbances.

The contributions of this paper are as follows: First, we design the modulation matrix for LPV-DS to address two types of disturbances. For small-scale disturbances, the modulation enhances the fitted accuracy of the generalization trajectory. For large-scale disturbances, it enables rapid convergence and regression toward the desired subtask space. Second, based on the Lyapunov stability analysis method, we prove that the proposed modulation is still globally asymptotically stable, ensuring smooth generalization of the whole complex task. Third, adaptive primitive segmentation is used to decompose complex tasks into simpler subtasks, which facilitates learning effectiveness and autonomously constructs a motion-level skill library. The skill decision engine will reconfigure skills based on task-level logic with modulation to resist disturbances. Finally, by conducting the LASA [47] and MNIST [48] handwriting dataset simulation and a sugar-scooping task, we demonstrate the effectiveness and robustness of our framework for complex skill transfer.

III. Preliminaries

A. Trajectory Segmentation

Considering multiple robot motion sequences for a complex task, our goal is to model the shared movement primitives between multiple sequences, reveal the correlation and segment them. Each motion primitive can be modeled individually by a parameter-independent Gaussian distribution or dynamic system, and the transitions between these movement primitives approximately follow Markovian. The HMM excels in handling time sequences and describing state transitions, which is quite popular for such Markov jump processes. Let $q_i = [q_{i1}, q_{i2}, \dots]$ be the feature set of the i-th demonstration trajectory, where each element takes on a value of either 1 or 0, denoting the presence or absence of a particular model, respectively. The HMM describes a time series consisting of the i-th trajectory $\{x_t^{(i)}\}_{t=0}^{T_i}$ and its corresponding sequence of hidden feature step $\{s_t^{(i)}\}_{t=0}^{T_i}$. Here, $s_t^{(i)} \in q_i$ and T_i denotes the length of the i-th trajectory.

The BP-AR-HMM is widely used for identifying shared features among multiple sequential series, which are considered a subset of behavioral patterns. However, when it comes to trajectory segmentation, the traditional switching vector autoregressive (VAR) process lacks spatial identification. It means that the trajectory segment with the same shape will be recognized as the same behavioral feature, regardless of their sequential series. In order to achieve segmentation of complex skill trajectories, we use Gaussian kernels as emission probability in HMM, while preserving the Beta process on the step transition. This approach is referred to as the Beta Process Gaussian Model HMM (BP-GM-HMM).

In particular, the Beta process obtains a non-parametric behavior set prior on the features from each demonstrated trajectory. Suppose we have N sequences, the Beta process H can be described as [49]

$$\begin{aligned} H|H_0, \alpha &\sim BP(\alpha, H_0) \\ Q_i|B &\sim BeP(H), \quad i = 1, \dots, N \\ Q_i &= \sum_k \omega_{ik} \delta_{\theta_k}, \end{aligned} \tag{1}$$

Here, H_0 denotes a finite base measurement of H and the concentration parameter α impacts the likelihood of adding a new feature. Q_i follows the Bernoulli process (BeP), implying that each feature indicator q_{ik} is drawn from a Bernoulli distribution with parameter ω_{ik}. Assuming we have N sequences, we can describe its posterior probability distribution as follows:

$$\begin{aligned} &H|Q_1, \dots, Q_N, H_0, \alpha \\ &\quad \sim BP(\alpha + N, \frac{\alpha}{\alpha + N} H_0 + \sum_{k=1}^{K} \frac{n_k}{\alpha + N} \delta_{\theta_k}), \end{aligned} \tag{2}$$

Here, K represents the total number of independent features across all sequences, and the k-th independent feature exists in n_k sequences. In fact, the above calculation process is typically performed using the Indian Buffet Process (IBP) to accomplish the sampling.

The transition parameter π_j defines the switching probability between hidden steps. For each feature set q_i, the transitions between features are defined as a sticky Dirichlet distribution:

$$\begin{aligned} s_t^{(i)} &\sim \pi^{(i)}_{s^{(i)}_{t-1}} \\ \pi_j^{(i)}|q_i, \beta, \kappa &\sim Dir([\beta, \dots, \beta, \beta + \kappa, \beta, \dots, \beta]), \end{aligned} \tag{3}$$

Here, the hyperparameters β and κ respectively define the divergence for Dirichlet distribution and the stickiness of the transition parameter.

For each hidden step variable s_t to observed variable y_t, we assume it follows a Gaussian distribution:

$$x_t^{(i)} \sim \mathcal{N}(\mu_{s_t^{(i)}}, \Sigma_{s_t^{(i)}}), \tag{4}$$

Among them, the mean $\mu_{s_t^{(i)}}$ and variance $\Sigma_{s_t^{(i)}}$ of a Gaussian distribution are modeled using the prior form of Normal-Inverse-Wishart (NIW). Moreover, the segmented subskills will be integrated to form a skill library. Here, we use the

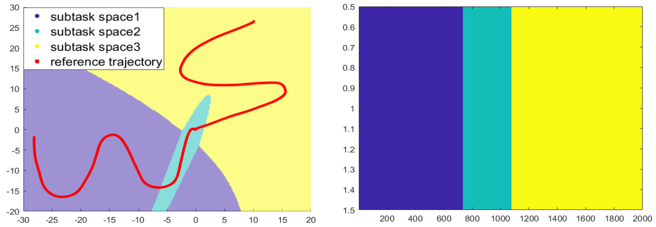

Fig. 2. An example of trajectory segmentation by BP-GM-HMM.

Gaussian model as the emission function for HMM because it captures spatial information effectively to cluster and achieves segmentation. Furthermore, it offers faster and more stable convergence.

B. Skill Decision Engine

Once a complex task is segmented into independent primitives, a semantic skill can be easily expressed as a deterministic sequence of these primitives. An example of the segmentation results from the demonstration is shown in Fig. 2. The trajectory is divided into 3 segments by different region boundaries. In the meanwhile, different feature models define the corresponding subtask spaces, which are distinguished by different colors. Behavior Tree (BT) is commonly used to organize primitives to accomplish specific semantic tasks, without requiring the user to understand the specific primitive forms. Instead, the user can focus on task-level programming, greatly reducing the difficulty and burden of programming. In our framework, we aim to shield the primitives of the underlying concrete form operators and encapsulate a task-level semantic programming interface.

As illustrated in Fig. 3, an example of a BT for a sugar-scooping task is depicted as a hierarchical tree structure with numerous nodes and branches, which describes the execution relationship between different primitives. Semantic programming includes switching relations and switching conditions, which often play a critical role in the smooth completion of tasks. The switching logic in the diagram includes *Sequences* and *Fallbacks*. The former necessitates the successful sequential execution of the corresponding subnodes, denoted by the symbol "→". The latter allows the task to be executed successfully through any of several feasible ways, denoted by the symbol "?". To automatically construct a BT, we first need to segment the motion sequences to obtain the subtask switching relations and conditions with BP-GM-HMM. Subsequently, the skill decision engine builds BT through these switching relations and conditions through the *Sequences* and *Fallbacks*. There are most common libraries available to implement BT [50]. The operator only needs to complete a demonstration of a skill, regardless of its complexity. During the generalization process, the BT will guide to select primitives for modulation and execution based on the actual manipulator state and task execution progress.

TABLE I
AN EXAMPLE OF TRAJECTORY SEGMENTATION

Time step	1	2	3	4	5	6	7	8
Sequence1	$x_1^{(1)}$	$x_2^{(1)}$	$x_3^{(1)}$	$x_4^{(1)}$	$x_5^{(1)}$	$x_6^{(1)}$	$x_7^{(1)}$	$x_8^{(1)}$
Sequence2	$x_1^{(2)}$	$x_2^{(2)}$	$x_3^{(2)}$	$x_4^{(2)}$	$x_5^{(2)}$	$x_6^{(2)}$	$x_7^{(2)}$	$x_8^{(2)}$

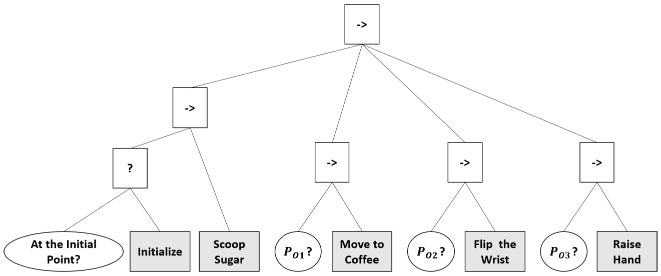

Fig. 3. An example of BT for a sugar-scooping task.

The BT will also automatically extract the appropriate switching conditions from multiple sequences. Assuming that the set of states belonging to step m segmented by (1) in the i-th sequence is denoted as $\mathcal{M}_m^{(i)}$, its end subscript can be further denoted as $T_m^{(i)}$. We define the set of primitive endpoints as $\{g_1, g_2, \ldots\}$ by averaging the endpoints of each segmented trajectory, namely $g_m = \Sigma_{i=1}^{N} x_{T_m^{(i)}}^{(i)}$. For example, in Table I, there are two trajectory sequences that are divided into three sets by BP-GM-HMM and marked with different colors. In order to obtain the attractor of each Dynamical System (DS), we define g_1, g_2, and g_3 as the average of $x_3^{(1)}$ and $x_2^{(2)}$, $x_6^{(1)}$ and $x_5^{(2)}$, and $x_8^{(1)}$ and $x_8^{(2)}$ respectively.

The main difference from a BT is that our decision engine not only selects subskills from the motion-skills library based on logical conditions but also modulates the expression of subskills based on the current step. For more details, please refer to Section IV.

C. Modeling of Skill Primitives

The convergence at the primitive level often serves as a protective mechanism for achieving smooth task completion. Therefore, it has a greater practical significance that a modeling method exhibits stability to further explore the enhancement of the fitting effect. Though suffering a disturbance, it ensures that every rollout in the generalization process can stably converge to the global unique target. This eliminates concerns about the robot getting stuck in a local step or transitioning unexpectedly to a state that may cause damage.

We employ LPV-DS constrained by P-QLF in trajectory modeling, aiming to obtain the following benefits (i) global asymptotic convergence; (ii) spatio-temporal robustness; (iii) nonlinear fitting; (iv) stable modulation. More precisely, it can be expressed in the following form [34]:

$$\dot{x} = f(x) = \sum_{k=1}^{K} \gamma_k(x)(A_k x + b_k) \tag{5}$$

with

$$\begin{cases} (A_k)^T P + P A_k = Q_k, \; Q_k = (Q_k)^T \prec 0 \\ b_k = -A_k g, \end{cases} \tag{6}$$

Here, A_k and b_k represent the parameters of the k-th linear system. P can be effectively solved using a non-linear positive semi-definite programming solver, such as PENLAB [51]. The auxiliary variable Q_k extends the search range of the parameter space, while g denotes the desired convergent state. The mixture coefficients $\gamma_k(x)$ and number of clusters K are learned through a pre-trained physically consistent Bayesian Gaussian mixture model, which incorporates an improved Dirichlet Process (DP) prior to better exploit the geometric properties of the demonstration data. The classified demonstration data will be used separately to train the parameters $\theta_m = \{A_k, b_k\}_{k=1}^{K}$ of each linear system via convex quadratic optimization:

$$\min_{\theta_m} J_m(\theta_m) = \sum_{i=1}^{N} \sum_{x_t^{(i)} \in \mathcal{M}_m^{(i)}} ||\dot{x}_t^{(i)} - \dot{x}_{ref,t}^{(i)}||, \tag{7}$$

where $\dot{x}_t^{(i)}$ and $\dot{x}_{ref,t}$ are derived from generalized velocity and the observed velocity in the training data respectively. By combining (5), (6), and (7), the parameter estimation of a primitive model can be completed. For each subtask, the generalization trajectory, the demonstration trajectory and the corresponding subtask space are shown in Fig. 4.

IV. Online Control Modulation Strategy

In a task generalization setting, each subskill should remain within an appropriate subtask space until reaching the primitive endpoint. In the sugar-scooping task, for instance, if the spoon performs the subskill of *Moving to Coffee* in the coffee cup, the cup is likely to be broken. The task generalization effect is most ideal when the initial state is near to coverage of the demonstration data. However, subtasks executed in unstructured environments are sensitive to disturbances, and the generalization effect will significantly reduce when outside the range of demonstration data. This can negatively impact the success rate of task execution and raise risks, especially in cases with strong disturbances.

In each subtask space, we modulate the control signal by embedding a multiplied matrix $M_m(x)$:

$$\dot{x} = \hat{f}_m(x) = M_m(x) f_m(x) \tag{8}$$

It provides an operable basis to improve the original trained LPV-DS and resist disturbances.

In BP-GM-HMM, the probability of a state belonging to a certain primitive can be calculated via (4). This enables a division of the entire space into subtask space $\mathcal{A}_m$ corresponding to segmentation before. Here, we will distinguish two types of disturbances based on whether the current state is within the corresponding subtask space $\mathcal{A}_m$. To further optimize the control signals outside the coverage of demonstration data, we construct the modulation matrix $M_m(x)$ for DS by performing singular value decomposition (SVD):

$$M_m(x) = \begin{cases} W_m(x) D_{m,ls}(x) W_m^{-1}(x), & p(x|\mathcal{A}_m) < p(x|\mathcal{A}_{m-}) \\ W_m(x) D_{m,ss}(x) W_m^{-1}(x), & p(x|\mathcal{A}_m) \geq p(x|\mathcal{A}_{m-}), \end{cases} \tag{9}$$

where A_{m-} denotes the primitive region other than $\mathcal{A}_m$. $W_m(x)$ denotes the matrix of eigenvectors. $D_{m,ls}(x)$ and $D_{m,ss}(x)$ are the matrices of eigenvalues for large-scale modulation and small-scale modulation, which will be designed in detail later. SVD enables us to individually control certain decoupled components, allowing for a more intuitive understanding and facilitating stability design. Furthermore, $W_m(x)$ can be expressed as:

$$W_m(x) = [e_{m,g}(x), e_{m,1}(x), \ldots, e_{m,d-1}(x)], \tag{10}$$

Here, $e_{m,1}(x) \ldots e_{m,d-1}(x)$ and $e_{m,g}(x)$ form an orthogonal matrix $W_m(x)$. To maintain stability easier, we have uniformly chosen a modulation direction:

$$e_{m,g}(x) = \frac{x - g_m}{||x - g_m||} \tag{11}$$

Afterward, we construct other eigenvectors using Schmidt orthogonalization so that each component in the diagonal matrix has a clear geometric meaning.

In the following, we will design the eigenvalue matrices for small-scale and large-scale disturbances respectively.

A. Large-Scale Disturbance Modulation

If the current state of the end-effector exceeds the relevant subtask space due to the disturbance, we consider the end-effector has experienced a large-scale disturbance. In this situation, both the environment and the end-effector itself will be at high risk since the control signals generated by the primitive can no longer be dependable. Therefore, it is crucial for the end-effector to promptly return to the appropriate subtask space. We construct the matrices of eigenvalues:

$$D_{m,ls}(x) = diag(\lambda_{m,ls}(x), 0, \ldots, 0), \tag{12}$$

While the modulation factor $\lambda_{m,ls}(x)$ in the target direction always retains the goal orientation, the components in the non-target direction are constrained in the equation above.

Assuming that due to large-scale disturbance, the end-effector is forced to move to another subtask space $\mathcal{A}_{m-}$ before the convergence of DS at step m, the modulation factor is designed as:

$$\lambda_{m,ls}(x) = -\mathrm{sgn}([\mathrm{W_m(x)^{-1}f(x)}]_1) \max\{\min\{\lambda_{\mathrm{limit}}, \frac{\mathrm{p(x|\mathcal{A}_{m-})}}{\mathrm{p(x|\mathcal{A}_m)}}\}, 1\} \tag{13}$$

Here, λ_{limit} denotes the maximum permitted acceleration multiplier. There is $p(x|\mathcal{A}_{m-}) > p(x|\mathcal{A}_m)$ and $1 < |\lambda_{m,ls}(x)| < \lambda_{limit}$ when the end-effector of the manipulator outside the expected subtask space. By enabling large-scale disturbance modulation, we accelerate end-effector regression to the appropriate subtask space. Obviously, $[W_m(x)^{-1} f(x)]_1$ represents the direction on the axis $e_{m,g}(x)$. It keeps the direction of $\dot{x}$ always pointing towards the target, thus ensuring convergence.

B. Small-Scale Disturbance Modulation

The convergence of our primitive's modeling assures that the influence of the generalized shape on the task success rate becomes progressively significant when the end-effector is located within its expected subtask space. However, the generated signals tend to have challenges in generalizing well when the demonstration data is insufficient, resulting in deviations in the shape of the generalized trajectory. In order to

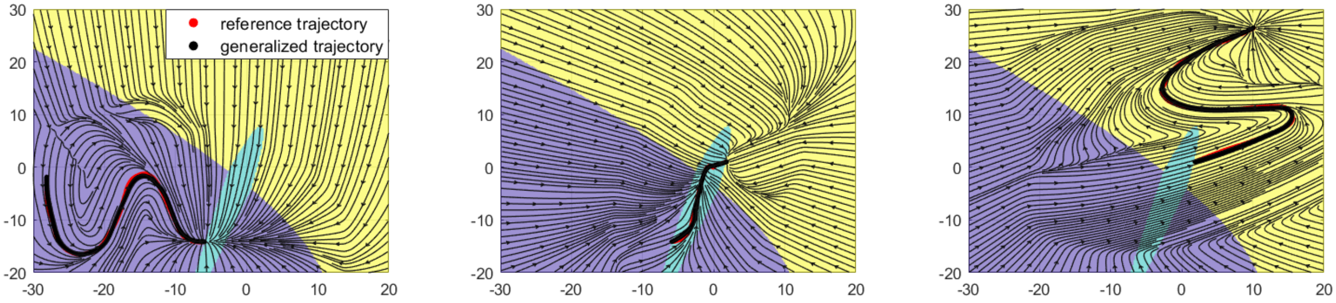

Fig. 4. GMM-based LPV-DS for modeling of trajectories.

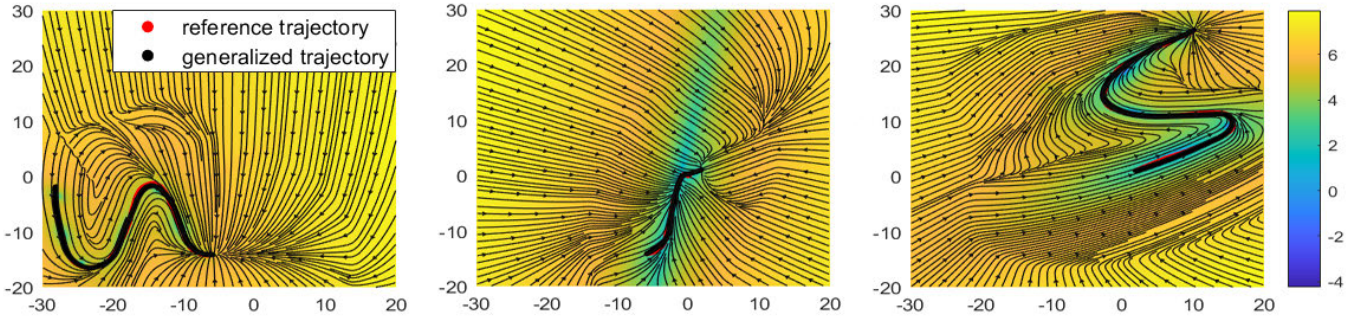

Fig. 5. Examples of the uncertainty colormaps calculated by Rényi entropy.

improve the similarities between the generalized shape and the demonstrated shape while still maintaining convergence, the subsequent control strategy aims to modulate the control signals in a new way. Thus, the modulation will be implemented with minimal impact on states that fall within the coverage of the demonstration data.

In Section III, before focusing on learning LPV-DS, it is necessary to first learn a GMM in order to obtain the mixture coefficients and cluster numbers mentioned earlier. Here, we further introduce quadratic Rényi entropy for primitives to calculate the uncertainty of the control signal:

$$H_2(p(u|x)) = -\log \mathbb{E}[\mathrm{p}^2(\mathrm{u}|\mathrm{x})], \tag{14}$$

Here $\mathbb{E}[\cdot]$ denotes mathematical expectation. Furthermore, Rényi entropy takes the following form for a GMM:

$$H_2(p(u|x)) = -\log \sum_{\mathrm{i}=1}^{\mathrm{K}} \sum_{\mathrm{j}=1}^{\mathrm{K}} \pi_{\mathrm{i}}(\mathrm{x})\pi_{\mathrm{j}}(\mathrm{x})\mathrm{e}^{\triangle_{\mathrm{ij}}}, \tag{15}$$

where

$$\triangle_{ij} = \frac{1}{2}\Big(\mu_{ij}^T \Sigma_{ij}^{-1} \mu_{ij} - (\mu_i^T \Sigma_i^{-1} \mu_i + \mu_j^T \Sigma_j^{-1} \mu_j) - \log \frac{|\Sigma_{\mathrm{i}}^{-1} + \Sigma_{\mathrm{j}}^{-1}|}{|\Sigma_{\mathrm{i}}^{-1}| + |\Sigma_{\mathrm{j}}^{-1}|} - \mathrm{d}\log 2\pi\Big), \tag{16}$$

Here, $\Sigma_{ij} = (\Sigma_i^{-1} + \Sigma_j^{-1})^{-1}$ and $\mu_{ij} = \Sigma_{ij}(\Sigma_i^{-1}\mu_i + \Sigma_j^{-1}\mu_j)$. $\triangle_{ij}$ reflects the exponential difference between the two clusters of GMM. The value of Rényi entropy is low in coverage of the demonstration data, indicating that the states there have less uncertainty, otherwise, the uncertainty is large. We visualize the uncertainty calculated by Rényi entropy as colormaps as shown in Fig. 5.

We believe that the insufficient generalization of the trajectory shape outside of the demonstrated region is caused by the non-target-oriented components of the control signal, which are not properly aligned with the target-oriented component. Therefore, we hope that the target-oriented component of the control signal remains intact while the non-target-oriented components of the control signal are attenuated. Moreover, the control signal should not push the end-effector away from the target and should have minimal impact on regions with low uncertainty. Based on the above principles, we can construct the matrix of eigenvalues:

$$D_{m,ss}(x) = \begin{cases} E, & H_2(x_r) \geq H_2(x) \\ D_{m,ls}(x) + \lambda_{m,ss}(x)E, & H_2(x_r) < H_2(x), \end{cases} \tag{17}$$

where

$$\lambda_{m,ss}(x) = min\{1, e^{H_2(x_r) - H_2(x)}\}, \tag{18}$$

Here, E represents the identity matrix and x_r is a reference state that can be chosen. Generally, we can choose the target state g_m as the reference state x_r. We set $\lambda_{m,ss}(x)$ to a maximum value of 1, meaning that no modulation is performed when the uncertainty of the current state is lower than the reference value $H_2(x_r)$.

$D_{m,ls}(x)$ is added in (17) for the following purpose: 1). In the expected subtask space, we have $\lambda_{m,ls}(x) = -\mathrm{sgn}([\mathrm{W}(\mathrm{x})^{-1}\mathrm{f}(\mathrm{x})]_1)$. When the signal deviates, $\lambda_{m,ls}(x) = -1$, while $\lambda_{m,ss}(x) < 1$ can guarantee that the first value of the diagonal matrix $D_{m,ss}$ is negative. 2). Since $\lambda_{m,ss}(x) < 1$, the control signal of the non-target-pointing components will be attenuated without $\lambda_{m,ls}(x)$. In the expected subtask space, we have $\lambda_{m,ls}(x) = 1$, which will accelerate the convergence process when moving to the target.

C. Proof of Stability

Subsequently, we use the Lyapunov stability theorem to verify the stability of the proposed modulation method. Considering a continuous and continuously differentiable Lyapunov candidate $V_m(x)$ of the m-th primitive in motion-skills library has the following form:

$$V_m(x) = \frac{1}{2}(x - g_m)^T P(x - g_m) \tag{19}$$

It's easy to prove that unmodulated $f_m(x)$ satisfies $\frac{\partial V_m(x)}{\partial x} f_m(x) < 0$ with $P \prec 0$. We just need to show that the modulated $\hat{f}_m(x)$ satisfies global asymptotic stability in a condition that the unmodulated $f_m(x)$ satisfies global asymptotic stability. When $p(x|\mathcal{A}_m) < p(x|\mathcal{A}_{m-})$,

$$\begin{aligned}
&\dot{V}_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} W_m(x) D_{m,ls}(x) W_m(x)^{-1} f_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} W_m(x) diag(\lambda_{m,ls}(x), 0, \ldots, 0) W_m(x)^{-1} f_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} e_{m,g}(x)^T \lambda_{m,ls}(x)[W_m(x)^{-1} f_m(x)]_1\\
&< 0,
\end{aligned} \tag{20}$$

When $p(x|\mathcal{A}_m) \geq p(x|\mathcal{A}_{m-})$ and $H_2(x_r) < H_2(x)$

$$\begin{aligned}
&\dot{V}_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} W_m(x) D_{m,ss}(x) W_m(x)^{-1} f_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} W_m(x)(D_{m,ls}(x) + \lambda_{m,ss}(x)E) W_m(x)^{-1} f_m(x)\\
&= \frac{\partial V_m(x)}{\partial x} W_m(x) D_{m,ls}(x) W_m(x)^{-1} f_m(x)\\
&\quad + \lambda_{m,ss}(x) \frac{\partial V_m(x)}{\partial x} f_m(x)\\
&< 0.
\end{aligned} \tag{21}$$

Therefore, our modulation $\hat{f}_m(x)$ is global asymptotic stability with respect to an attractor g_m.

V. Experiments

A. Simulation

We validate the proposed imitation learning framework with the LASA handwriting dataset and perform a visual comparative analysis using a group called Multi_Models_3 as a complex task. Only one trajectory is used for training and the remaining six trajectories are used to test the generalization accuracy.

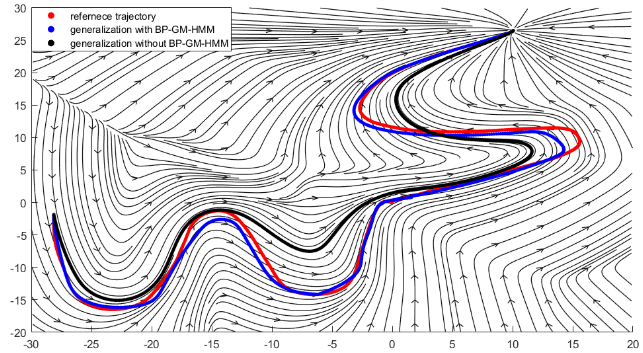

Fig. 6. The performance of LPV-DS on different segmentations.

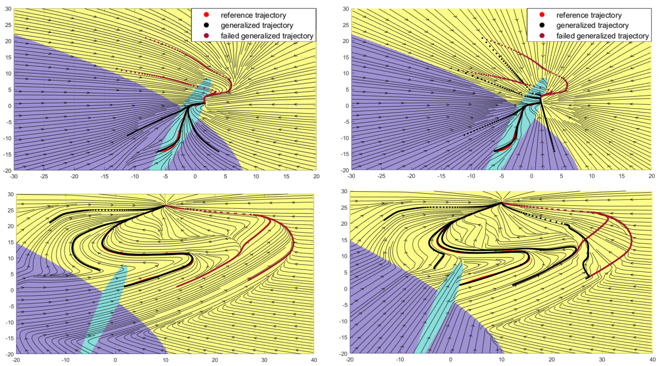

Fig. 7. Modulation comparisons. The left column represents the generalized shape before modulation, while the right column represents the shape after modulation. The top two images represent large-scale disturbances, while the bottom two represent small-scale disturbances.

Fig. 6 illustrates how BP-GM-HMM improves the generalizability. We compare the rollouts before and after segmentation, and it is obvious that the rollout processed by segmentation is more similar in shape and spatial position to the reference trajectory. Increasing the number of segments greatly improves the generalization accuracy of task learning, which reduces the complexity requirement of the primitive model and thus improves the fitting accuracy of each model. However, excessive segments will lead to an increment of the decision-making difficulty of the skill decision engine, potentially resulting in a limited field of view that is harmful when dealing with large-scale disturbances. It is worth noting that during the segmentation of a complex trajectory, the semantically prior and the number of segments are not required. Besides, BP-GM-HMM not only segments the trajectory, but also segments the entire 2D spaces, which can be used as the latent subtask spaces.

Fig. 7 illustrates the generalization effect before and after modulation, which enhances the performance of the trajectory convergence. Since the second subtask space is small and the third subtask space is large, we simulate the resistance effect of the control modulation strategy on large-scale disturbances and small-scale disturbances in each of the two stages with 5 initial configurations. Large-scale disturbances are simulated as generalizing from some initial position outside the current subtask space. The rollouts with two initial configurations before modulation (red trajectories) return to the subtask space through a redundant and tortuous path, which is dangerous and unreliable for the end-effector. The modulated rollouts will quickly converge to the task subspace as we expected. When the end-effector returns to the current subtask space, the fast convergence modulation strategy will be disabled. Similarly,

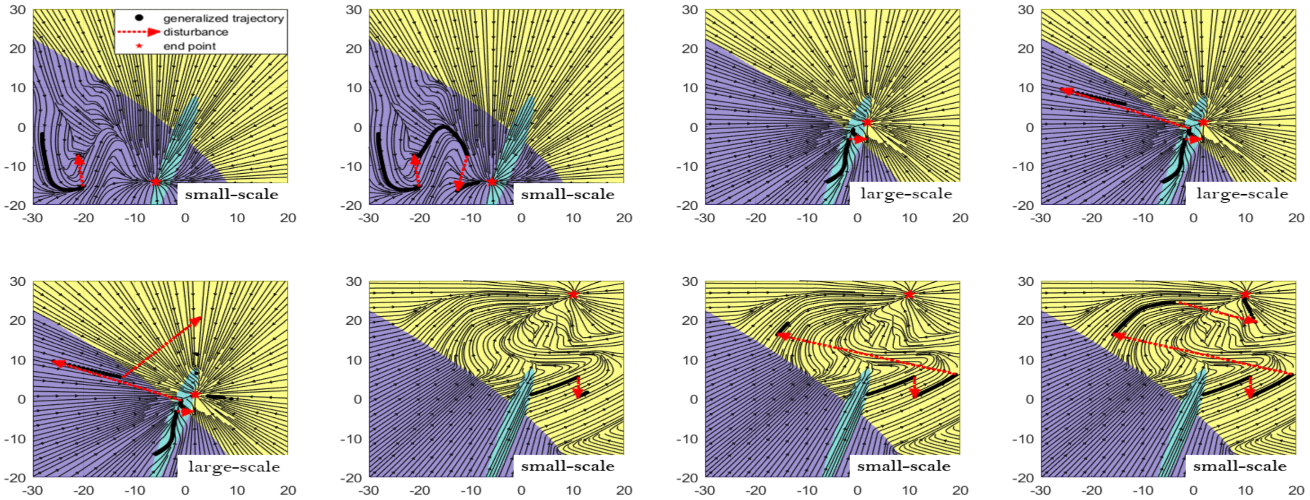

Fig. 8. Rollouts under both large-scale and small-scale disturbances.

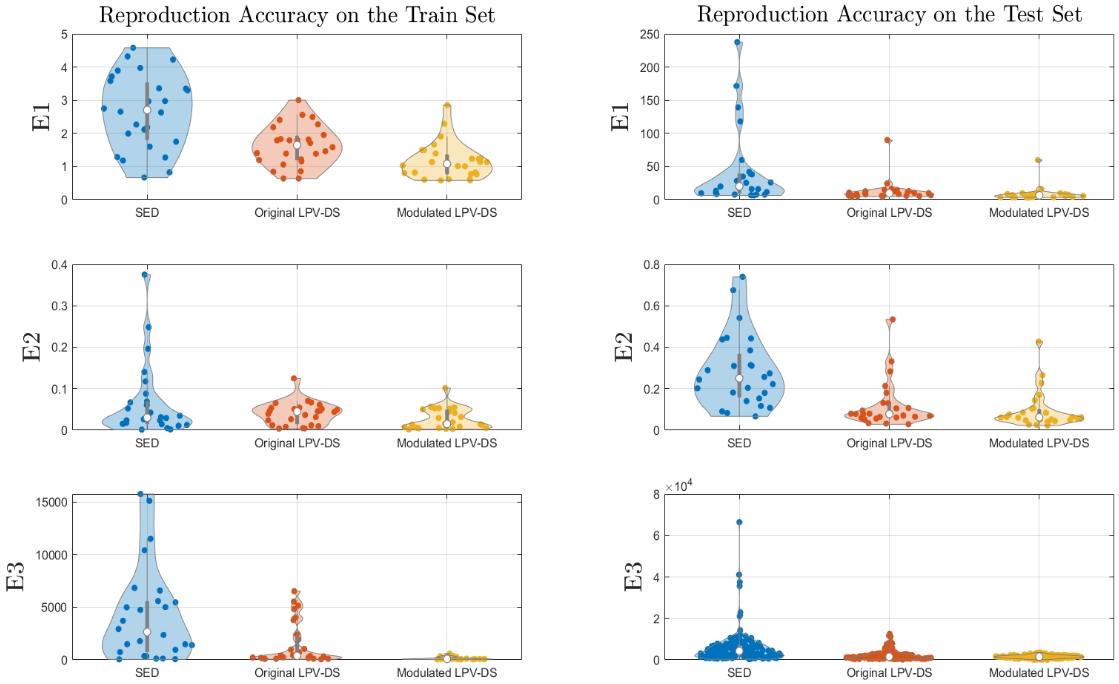

Fig. 9. Comparison results on the LASA dataset.

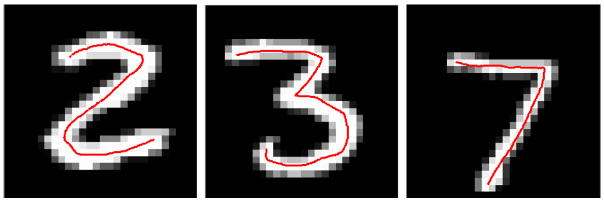

Fig. 10. Examples of the trajectories drawn from the MNIST digit images.

small-scale disturbances are simulated as generalizing from some initial position in the current task subspace but away from the demonstration range. The shape of the rollouts under the two initial configurations before modulation (red trajectories) is quite different from the shape of the training trajectories. But after modulation, the shape of the two rollouts is maintained and the rollouts all try to return to the demonstration range. It is worth noting that since we set a threshold on the Rényi entropy, the good rollout within the demonstration range is not affected, which meets the expectation of our proposed method.

In order to reveal the influence of the skill decision engine on the handwriting task, we try to simulate a complete generalization process as shown in Fig. 8. It is discretely disturbed but exhibits a high robustness. The skill decision engine determines a suitable subtask space according to the current and previous steps, and our modulation strategy will be carried out based on the current subtask space. Switching is not performed until it converges to the target state in the current

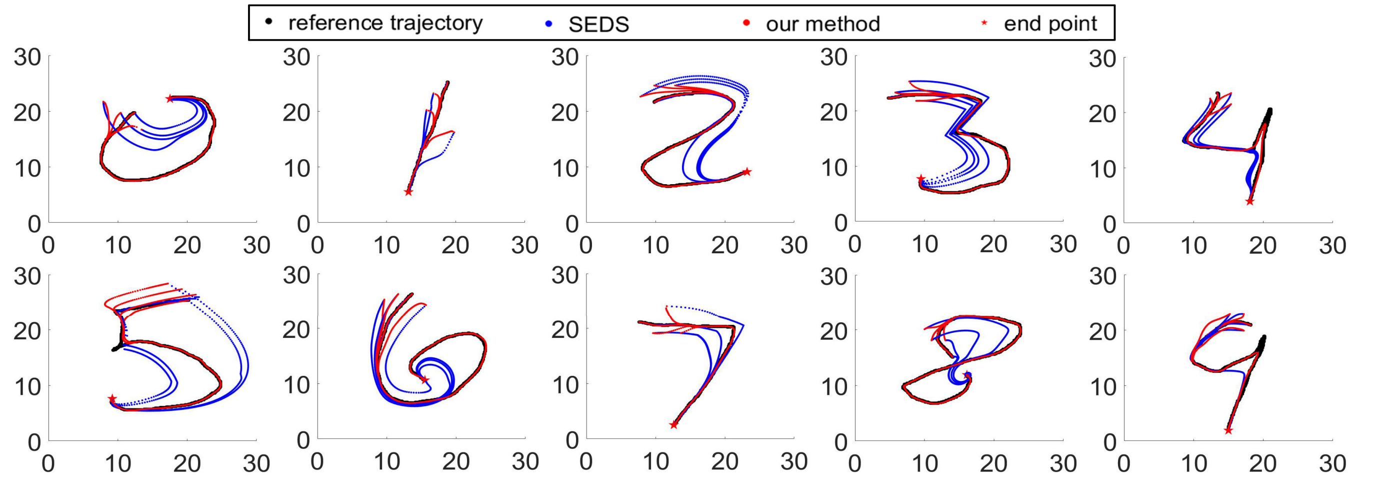

Fig. 11. Comparison of generalization trajectories for the MNIST handwritten digit sequences.

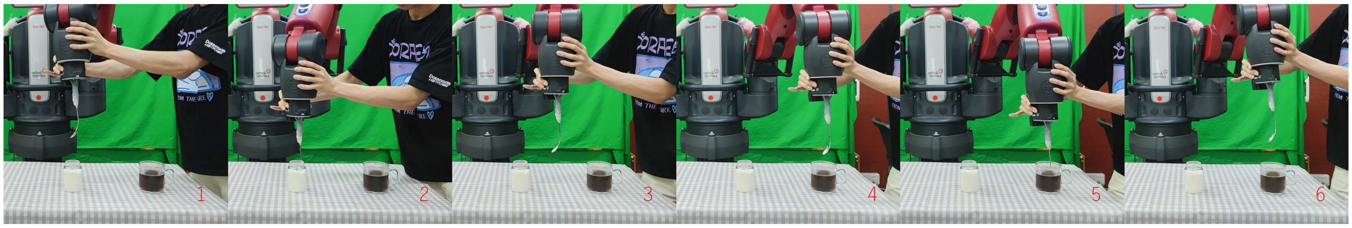

Fig. 12. Demonstration process for the sugar-scooping task.

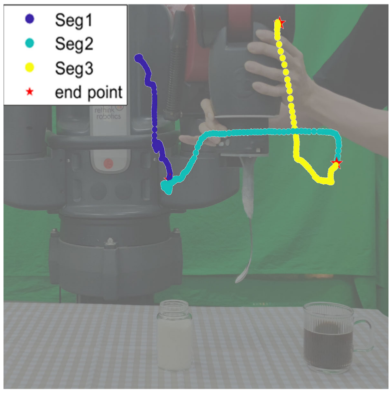

Fig. 13. Segmentation by BP-GM-HMM for the sugar-scooping task.

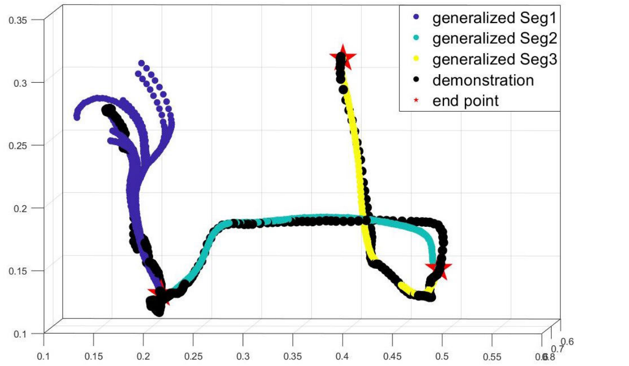

Fig. 14. The generalized position trajectories under 7 configurations for the sugar-scooping task.

subtask space. The disturbances are performed randomly in position, direction and magnitude. Among them, the fourth and fifth subfigures depict large-scale disturbances, where the end-effector is moved out of the central green subtask space. The remaining figures depict small-scale disturbances, where the end-effector is disturbed but still remains within the corresponding subtask space.

We employ three indexes for evaluating generalization errors quantitatively:

$$E1 = \frac{1}{|S|T_i}\sum_{i\in S}\sum_{t=1}^{T_i}||\dot{x}_{ref,t}^{(i)} - M(x_{ref,t}^{(i)})f(x_{ref,t}^{(i)})||$$

$$E2 = \frac{1}{|S|T_i}\sum_{i\in S}\sum_{t=1}^{T_i}|1 - \frac{\dot{x}_{ref,t}^{(i)T}M(x_{ref,t}^{(i)})f(x_{ref,t}^{(i)})}{||\dot{x}_{ref,t}^{(i)}||\cdot||M(x_{ref,t}^{(i)})f(x_{ref,t}^{(i)})||}|$$

$$E3 = \frac{1}{|S|T_i}\sum_{i\in S}\sum_{t=1}^{T_i}\mathcal{A}(x_{ref,t}^{(i)}, x_{ref,t+1}^{(i)}, x_t^{(i)}, x_{t+1}^{(i)}),$$

where, $E1$, $E2$, and $E3$ are the root mean square error, cosine similarity and dynamic time warping distance. S denotes a subset of demonstration, and $\mathcal{A}(\cdot)$ denotes the quadrilateral region enclosed by four vertices. Here we employ BP-GM-HMM for segmentation and subsequently train the models to obtain a skill library. To compare the generalization accuracy, we evaluate SED, LPV-DS without modulation, and LPV-DS after modulation across all 26 samples in the LASA dataset. The results of this comparison are shown in Fig. 9. Violin plots reveal intuitive distributions of various methods, as well as mean and variance. It's easy to find that our modulation

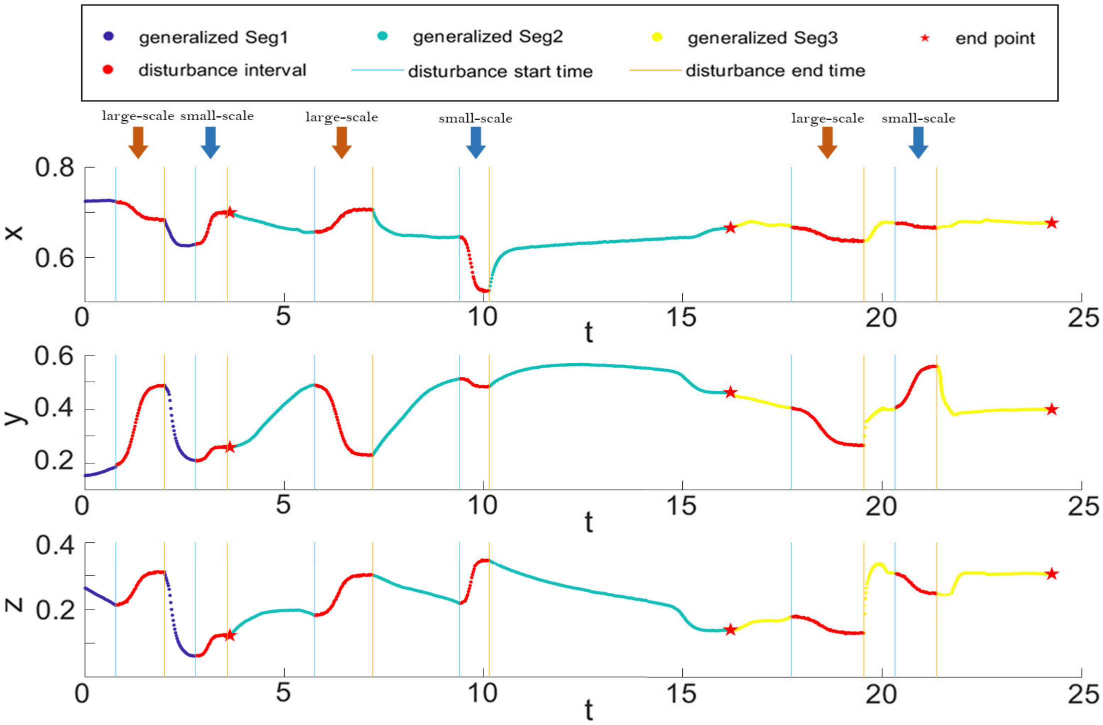

Fig. 15. The generalized position trajectories of the robot end-effector under disturbances in the sugar-scooping task. It shows that the robot can coverage to the desired points (red stars) with the proposed method under both small-scale and large-scale disturbances (red lines).

exhibits lower error mean and variance in all three error measurements, both for train and test sets.

We further verify our imitation framework on the MNIST dataset. MINST is a handwriting image data set mainly used for image classification. Since the demonstration data in imitation learning typically takes the form of trajectory sequences, we need to first extract the trajectories of the digits in the MNIST dataset, which will be further used for model training. Examples are given in Fig. 10. To clearly and intuitively illustrate the generalization performance of our method, Fig. 11 shows the generalization trajectories at different initial positions, with a comparison with the SEDS method. It clearly shows that the generalized trajectories using our method are almost consistent with the reference trajectories. Despite encountering sharp returns in digits 4 and 9, our method still achieves good fitting performance and outperforms SEDS.

B. The Sugar-Scooping Task

We validate the practicality of the proposed imitation learning framework on a Baxter robot. The main computations of our framework are implemented by MATLAB, while simultaneously receiving sensor information from Baxter and transmitting control signals through UDP communication. The robot is controlled under the PD-based impedance controller.

Here we conduct the coffee scooping task with multiple steps and complex trajectories. The experimental setup and a kinesthetic teaching process are shown in Fig. 12. The end-effector of Baxter's left arm is equipped with a spoon, which starts from the initial area, approaches the sugar can, scoops up a spoon of sugar, moves towards the coffee cup, and then lifts it away from the cup. We demonstrate it only once thus obtaining a limited demonstration dataset for model training. Following the denoising of the trajectories through window smoothing filtering, the training trajectories were adaptively divided into 3 segments using BP-GM-HMM without any prior knowledge, each distinguished by various colors as shown in Fig. 13. Notably, it is apparent to find that these segments exhibit strong semantically meaningful characteristics, nearly aligning with the process we described. Benefiting from Gaussian emission, there are consistent segmentation results for the demonstration trajectory.

Next, we use each segment to train a P-QLF-constrained LPV-DS as a skill primitive, and the skill decision engine selects an appropriate skill to execute from the skill library according to the behavior tree in Fig. 3 and the current step of the manipulator. We evaluate the performance of the generalized complete task trajectories under all 7 configurations. During this evaluation, sub-skills are switched upon convergence to the end of each segment. The obtained results are shown in Fig. 14. In these configurations, the trajectories are successfully generalized and they are almost identical to the demonstration trajectories and successfully scoop sugar into coffee, maintaining successful generalization at both the motion and task level. Without the stable convergence of each motion primitive, there is no suitable way to switch skills, and our sugar-scooping task cannot be completed.

Finally, we validate the robustness of the proposed imitation learning framework. Applying disturbances through manual traction is easily achievable with the Baxter. We conducted 50 generalization experiments, 48 of which completed the task, with a success rate of 96%. Specifically, we apply both large-scale disturbances, which represent task-level, and small-scale disturbances, which represent motion-level. The results of these simulations are illustrated in Fig. 15. We apply two types of disturbances in all three stages, demonstrating the robustness of the framework we propose in the practical scene.

VI. Conclusion

We propose a novel robust imitation learning framework for complex tasks, consisting of three main aspects: preprocessing, learning, and generalization. During the generalization process, we have designed modulation strategies to address both the convergence recovery from large-scale disturbances and the fitting accuracy for small-scale disturbances. As we know, today imitation learning has been trying to find a balance between fitting accuracy and robustness, and we aim to reorganize the design intention of our proposed framework from these two perspectives rather than the logical order of learning and generalization processing described before.

Our method has shown an improved fitting accuracy. First, adaptive segmentation reduces the fitting difficulty of single-model learning. Second, the introduction of P-QLF constraints alleviates the fitting inadequacies in SEDS to a certain extent. Lastly, the skill decision engine incorporates modulation strategies specifically tailored for small-scale disturbances, ensuring better coordination of control variables across target-oriented and non-target-oriented components.

Our method has also shown good robustness. First, the LPV-DS, serving as a state-driven modeling approach, provides excellent spatio-temporal robustness and stability for generalization. Even when the state transitions to other regions, it effectively controls the generalization process. Second, the skill decision engine incorporates modulation strategies specifically designed for large-scale disturbances. When the end-effector unexpectedly moves into other subtask spaces, the modulation strategy accelerates the recovery process, bringing the end-effector back to the relevant subtask space. Lastly, the behavior tree within the skill decision engine ensures task-level robustness.

Our method has two limitations. First, we find that the modulation will sometimes reduce the fitting accuracy to maintain global asymptotic stability, failing to fully restore the system to its state before encountering disturbances. Additionally, the framework lacks the ability for online relearning, rendering it will fail when existing modulation strategies need to be modified. In future work, we will explore the design of modulation strategies from the perspective of local state recovery. Additionally, we will incorporate online learning capabilities and autonomously construct behavior trees within the framework.

References

[1] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating human–robot collaborative tasks by teaching-learning-collaboration from human demonstrations," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 640–653, Apr. 2019.

[2] H. Wu, W. Yan, Z. Xu, T. Cheng, and X. Zhou, "A framework of robot skill learning from complex and long-horizon tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3628–3638, Oct. 2022.

[3] W. Si, N. Wang, and C. Yang, "A review on manipulation skill acquisition through teleoperation-based learning from demonstration," *Cognit. Comput. Syst.*, vol. 3, no. 1, pp. 1–16, Mar. 2021.

[4] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C.-Y. Su, "Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 329–340, Jan. 2018.

[5] C. Zeng, C. Yang, Q. Li, and S. Dai, "Research progress on human–robot skill transfer," *Acta Automatica Sinica*, vol. 45, no. 10, pp. 1813–1828, 2019.

[6] Y. Du, J. Jian, Z. Zhu, D. Pan, D. Liu, and X. Tian, "A syntactic method for robot imitation learning of complex sequence task," *Robotic Intell. Autom.*, vol. 43, no. 2, pp. 132–143, May 2023.

[7] S. Schaal, "Dynamic movement primitives—A framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*. Berlin, Germany: Springer, 2006, pp. 261–280.

[8] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *Int. J. Robot. Res.*, vol. 42, no. 13, pp. 1133–1184, Nov. 2023.

[9] C. Paxton, N. Ratliff, C. Eppner, and D. Fox, "Representing robot task plans as robust logical-dynamical systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5588–5595.

[10] D. H. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 261–266.

[11] S. Niekum, S. Chitta, A. G. Barto, B. Marthi, and S. Osentoski, "Incremental semantically grounded learning from demonstration," in *Proc. Robot., Sci. Syst.*, vol. 9, Berlin, Germany, 2013, p. 15607.

[12] N. B. Figueroa Fernandez, "From high-level to low-level robot learning of complex tasks: Leveraging priors, metrics and dynamical systems," EPFL, Lausanne, Tech. Rep., 2019. [Online]. Available: https://doi.org/10.5075/epfl-thesis-9631

[13] H. Kim, C. Oh, I. Jang, S. Park, H. Seo, and H. J. Kim, "Learning and generalizing cooperative manipulation skills using parametric dynamic movement primitives," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 3968–3979, Oct. 2022.

[14] S. Soheily-Khah and P.-F. Marteau, "Sparsification of the alignment path search space in dynamic time warping," *Appl. Soft Comput.*, vol. 78, pp. 630–640, May 2019.

[15] C.-H. Lee, C.-R. Lin, and M.-S. Chen, "Sliding window filtering: An efficient method for incremental mining on a time-variant database," *Inf. Syst.*, vol. 30, no. 3, pp. 227–244, May 2005.

[16] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst., Man Cybern., B*, vol. 37, no. 2, pp. 286–298, Apr. 2007.

[17] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, Feb. 2013.

[18] D.-H. Zhai, Z. Xia, H. Wu, and Y. Xia, "A motion planning method for robots based on DMPs and modified obstacle-avoiding algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 2678–2688, Oct. 2023.

[19] A. Gams, B. Nemec, A. J. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 816–830, Aug. 2014.

[20] C. Zeng, Y. Li, J. Guo, Z. Huang, N. Wang, and C. Yang, "A unified parametric representation for robotic compliant skills with adaptation of impedance and force," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 2, pp. 623–633, Apr. 2022.

[21] C. Zeng, H. Su, Y. Li, J. Guo, and C. Yang, "An approach for robotic leaning inspired by biomimetic adaptive control," *IEEE Trans. Ind. Informat.*, vol. 18, no. 3, pp. 1479–1488, Mar. 2022.

[22] J. Li, M. Cong, D. Liu, and Y. Du, "Enhanced task parameterized dynamic movement primitives by GMM to solve manipulation tasks," *Robotic Intell. Autom.*, vol. 43, no. 2, pp. 85–95, May 2023.

[23] L.-H. Kong, W. He, W.-S. Chen, H. Zhang, and Y.-N. Wang, "Dynamic movement primitives based robot skills learning," *Mach. Intell. Res.*, vol. 20, no. 3, pp. 396–407, Jun. 2023.

[24] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Auto. Robots*, vol. 42, no. 3, pp. 529–551, Mar. 2018.

[25] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *Int. J. Robot. Res.*, vol. 38, no. 7, pp. 833–852, Jun. 2019.

[26] Z. Jin, A. Liu, W.-A. Zhang, L. Yu, and C. Yang, "Gaussian process movement primitive," *Automatica*, vol. 155, Sep. 2023, Art. no. 111120.

[27] A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach*. Cambridge, MA, USA: MIT Press, 2022.

[28] W. Lohmiller and J.-J.-E. Slotine, “On contraction analysis for non-linear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, Jun. 1998.

[29] H. K. Khalil, *Nonlinear Control*, vol. 406. New York, NY, USA: Pearson, 2015.

[30] S. Singh, S. M. Richards, V. Sindhwani, J.-J.-E. Slotine, and M. Pavone, “Learning stabilizable nonlinear dynamics with contraction-based regularization,” *Int. J. Robot. Res.*, vol. 40, nos. 10–11, pp. 1123–1150, Sep. 2021.

[31] S. M. Khansari-Zadeh and A. Billard, “Learning stable nonlinear dynamical systems with Gaussian mixture models,” *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 943–957, Oct. 2011.

[32] K. Neumann and J. J. Steil, “Learning robot motions with stable dynamical systems under diffeomorphic transformations,” *Robot. Auto. Syst.*, vol. 70, pp. 1–15, Aug. 2015.

[33] C. Blocher, M. Saveriano, and D. Lee, “Learning stable dynamical systems using contraction theory,” in *Proc. 14th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Jun. 2017, pp. 124–129.

[34] N. B. F. Fernandez and A. Billard, “A physically-consistent Bayesian non-parametric mixture model for dynamical system learning,” in *Proc. Mach. Learn. Res.*, 2018, pp. 1–20.

[35] S. Mohammad Khansari-Zadeh and A. Billard, “Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions,” *Robot. Auto. Syst.*, vol. 62, no. 6, pp. 752–765, Jun. 2014.

[36] R. Caccavale, M. Saveriano, A. Finzi, and D. Lee, “Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction,” *Auto. Robots*, vol. 43, no. 6, pp. 1291–1307, Aug. 2019.

[37] Y. Wang, N. Figueroa, S. Li, A. Shah, and J. Shah, “Temporal logic imitation: Learning plan-satisficing motion policies from demonstrations,” 2022, *arXiv:2206.04632*.

[38] K. R. Dixon and P. K. Khosla, “Trajectory representation using sequenced linear dynamical systems,” in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, Apr. 2004, pp. 3925–3930.

[39] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, “Incremental local online Gaussian mixture regression for imitation learning of multiple tasks,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 267–274.

[40] D. Kulic, W. Takano, and Y. Nakamura, “Online segmentation and clustering from continuous observation of whole body motions,” *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 1158–1166, Oct. 2009.

[41] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, “Learning grounded finite-state representations from unstructured demonstrations,” *Int. J. Robot. Res.*, vol. 34, no. 2, pp. 131–157, Feb. 2015.

[42] C. Yang, C. Zeng, Y. Cong, N. Wang, and M. Wang, “A learning framework of adaptive manipulative skills from human to robot,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1153–1161, Feb. 2019.

[43] A. Wahrburg, S. Zeiss, B. Matthias, J. Peters, and H. Ding, “Combined pose-wrench and state machine representation for modeling robotic assembly skills,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 852–857.

[44] M. Colledanchise and P. Ögren, “How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees,” *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 372–389, Apr. 2017.

[45] M. Guo, K. H. Johansson, and D. V. Dimarogonas, “Revising motion planning under linear temporal logic specifications in partially known workspaces,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 5025–5032.

[46] S. Vougioukas, “Annual review of control, robotics, and autonomous systems,” *Agricult. Robot.*, vol. 2, no. 1, pp. 365–392, 2019.

[47] S. K.-Z. A. Billard. (2015). *Handwriting Human Motion Dataset*. [Online]. Available: https://github.com/justagist/pyLasaDataset

[48] Y. LeCun, C. Cortes, and C. J. C. Burgesh. (1994). *The Mnist Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[49] E. B. Fox, M. C. Hughes, E. B. Sudderth, and M. I. Jordan, “Joint modeling of multiple time series via the beta process with application to motion capture segmentation,” *Ann. Appl. Statist.*, vol. 8, no. 3, pp. 1281–1313, Sep. 2014, doi: 10.1214/14-aoas742.

[50] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, “A survey of behavior trees in robotics and AI,” *Robot. Auto. Syst.*, vol. 154, Aug. 2022, Art. no. 104096.

[51] J. Fiala, M. Kočvara, and M. Stingl, “Penlab: A MATLAB solver for nonlinear semidefinite optimization,” 2013, *arXiv:1311.5240*.

Weiyong Wang received the B.S. degree in automation control from the School of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2021, where he is currently pursuing the master’s degree. His current research interests include robotics, intelligent control, and imitation learning.

Chao Zeng (Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the South China University of Technology, Guangzhou, China, in 2019. He visited the Department of Informatics, University of Hamburg (UHH), Hamburg, Germany, from 2018 to 2019. He was a Post-Doctoral Research Fellow with UHH from 2020 to 2023. He is currently a Research Fellow with the University of Liverpool. His research interests include robot learning and control and physical human–robot interaction.

Hong Zhan received the M.S. degree in automation from the South China University of Technology, Guangzhou, China, in 2012, where she is currently pursuing the Ph.D. degree with the School of Automation Science and Engineering. She works with the School of Automation Science and Engineering, South China University of Technology. Her research interests include robotics, intelligent control, and human–robot interaction.

Chenguang Yang (Fellow, IEEE) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University, Xi’an, China, in 2005, and the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010. He performed postdoctoral studies in human robotics with Imperial College London, London, U.K., from 2009 to 2010. He is currently the Chair of Robotics with the Department of Computer Science, University of Liverpool, U.K. His research interests include human–robot interaction and intelligent system design. He was awarded the U.K. EPSRC UKRI Innovation Fellowship and the Individual EU Marie Curie International Incoming Fellowship. As the lead author, he won IEEE TRANSACTIONS ON ROBOTICS Best Paper Award in 2012 and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS Outstanding Paper Award in 2022. He is the Corresponding Co-Chair of the IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing.