

# Feature Space Decomposition for Effective Robot Adaptation

Chi Zhang<sup>1</sup>, Hao Zhang<sup>2</sup>, and Lynne E. Parker<sup>1</sup>

**Abstract**—Adaptation is an essential capability for intelligent robots to work in new environments. In the learning framework of Programming by Demonstration (PbD) and Reinforcement Learning (RL), a robot usually learns skills from a latent feature space obtained by dimension reduction techniques. Because the latent space is optimized for a specific environment during the training phase, it typically contains fewer variations. Accordingly, searching for a solution within the latent space can be less effective for robot adaptation to new environments with unseen changes. In this paper, we propose a novel Feature Space Decomposition (FSD) approach to effectively address the robot adaptation problem, which is directly applicable to the learning framework based on PbD and RL. Our FSD method decomposes the high-dimensional original features extracted from the demonstration data into *principal* and *non-principal* feature space. Then, the non-principal features are used to form a new low-dimensional search space for autonomous robot adaptation based on RL, which is initialized using a generalized trajectory represented by a Gaussian Mixture Model that is learned from the principal features. The scalability of our FSD approach guarantees that optimal solutions can be found in the new non-principal space, if they exist in the original feature space. Experimental results on real robots validate that our FSD approach enables the robots to effectively adapt to new environments, and is usually able to find optimal solutions more quickly than traditional approaches when significant environment changes occur.

## I. INTRODUCTION

When robots operate in real-world, dynamic environments, they must have the critical capability of adapting the learned knowledge to new scenarios never experienced in the training phase. Particularly, in Programming by Demonstration (PbD) [1], which enables a robot to learn a task by building a state-action mapping from human demonstrations, *adaptability to unseen situations* is a crucial research problem that must be addressed when the robot encounters a state with no prior demonstrations [2]. Reinforcement Learning (RL) is a widely used methodology to deal with this problem, which allows a robot to update its knowledge through autonomously interacting with the environment [3], [4]. However, RL often suffers from the curse of dimensionality when applied on robots with a high Degree of Freedom (DoF) [3], [4], [5]. Dimension Reduction (DR) [5], [6], [7] is usually applied to tackle this issue by projecting the original high-dimensional learning space to a low-dimensional latent space. However,

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1427004.

<sup>1</sup>Chi Zhang and Lynne E. Parker are with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA. {czhang24, leparker}@utk.edu

<sup>2</sup>Hao Zhang is with the HCRobotics Lab in the Department of Electrical Engineering and Computer Science, Colorado School of Mines, Golden, CO 80401, USA. hzhang@mines.edu

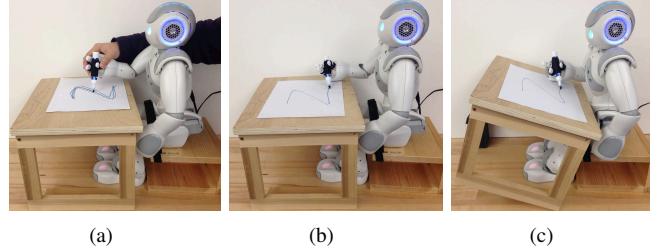


Fig. 1. A motivating example of robot adaptation. To teach a writing task, a human instructor physically moves the arm of the NAO robot (Fig. 1(a)). The sequence of robot body configurations (e.g., joint angles) are recorded as demonstrations for robot learning, which are encoded as features in the latent space. In an unchanged environment (Fig. 1(b)), the robot can successfully perform the task using the learned knowledge. In new environments (e.g., a tilted table top as in Fig. 1(c)), robot adaptation is necessary. (An accompanying video demonstrates robot adaptation on two tasks using the proposed approach. <https://youtu.be/W10DbmLhadI>)

there is no theoretical guarantee that the optimal policy lies in the *latent space* [6]. In extreme cases, such approaches may not be able to find any solution, when all policies, including sub-optimal ones, fall outside the latent space [6], [7].

Fig. 1 demonstrates a motivating example of a writing task performed by a NAO robot. When a human instructor teaches the robot by physically moving its arm, sensory data, such as absolute joint angles and end-effector positions with respect to the robot torso, are recorded. However, information related to the environment, such as the setup of the table (Fig. 1(b)), is usually not explicitly provided and/or is often ignored by human instructors. In this scenario, the robot only learns from its own configurations (e.g., joint positions) within a specific environment that is not well represented. When PbD and RL are applied to learn the writing task, the latent space is tuned and optimized to include features that are most suitable for the (unrepresented) training environment. In other words, the latent space is restricted by the environment during training, which typically cannot be fully represented. When the NAO robot encounters new situations (e.g., tilted or tilted table top as illustrated in Fig. 1(c)), the learned knowledge is no longer directly applicable and robot adaptation is required. Traditional RL techniques [6], [7] only explore the latent space, which is optimized for the specific environment during the training phase and typically contains fewer variations. As a result, the adaptation approaches based on the latent feature space can be ineffective, since optimal solutions for new environments may not exist in the restricted latent space that is intentionally optimized for the training environment.

In this paper, we propose a novel *Feature Space Decomposition* (FSD) approach that is directly applicable to the

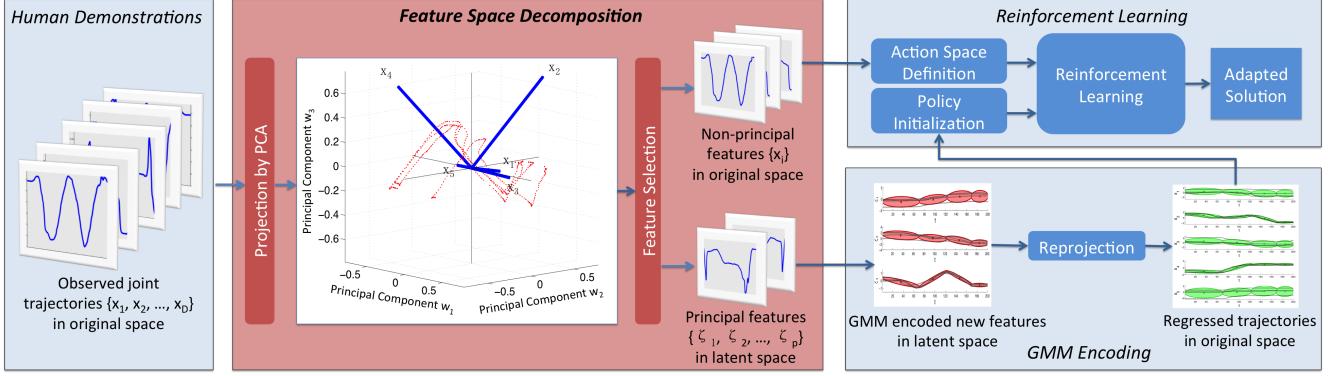


Fig. 2. Overview of the complete robot adaptation system, with our new FSD approach highlighted in red.

PbD and RL learning framework to effectively address the autonomous robot adaptation problem. Our FSD approach divides all features from human demonstration data into two categories: *principal* and *non-principal*. Principal features are the ones optimized to represent the human demonstration data in the environment during training, which are equivalent to the features forming the latent space in traditional RL approaches. Non-principal features encode the information that is not optimized for the training environments, which usually contains more variations and is less restricted by the training process. Fig. 2 illustrates how our FSD method can be directly applied in the PbD and RL learning framework to construct a complete autonomous robot adaptation system. Similar to traditional PbD approaches, the principal features are employed to train a Gaussian Mixture Model (GMM), and then reprojected onto the original feature space to obtain a generalized trajectory, which is applied to initialize RL for robot adaptation. Different from previous RL approaches [6], [7] that typically use the latent space only, to discover solutions for new situations, our adaptation system explores the non-principal feature space, which is less restricted by the training process and has more variations.

Our contributions are twofold. First, we introduce a new perspective of autonomous robot adaptation within the PbD and RL learning framework, i.e., learning from outside the restricted latent space. This perspective can be especially beneficial, if new situations are caused by significant changes of the environment when solutions may not exist in the latent space (that is optimized for training environments). Second, we propose a simple, yet effective FSD approach to realize our perspective for robot adaptation. The method is effective since it can be scaled to the entire feature space and thus is guaranteed to find an optimal solution even when it does not exist in the latent space. When new situations are caused by significant environment changes, FSD-based RL approaches can usually find solutions more quickly.

The remainder of the paper is organized as follows. After reviewing related work in Section II, we discuss our feature space decomposition approach in Section III. Experimental results are presented in Section IV. Finally, we conclude this paper in Section V.

## II. RELATED WORK

A large number of PbD methods are based on probabilistic models [2]. Inamura et al. [8] proposed the Mimesis Model in which trajectories are encoded by a Hidden Markov Model (HMM) and then multiple HMMs are used to retrieve new generalized motions using a stochastic process. Vakanski et al. [9] extended this model by encoding demonstrated trajectories with HMM and generating a generalized trajectory using key points. GMM is another well known approach used by Calinon et al. [10] to retrieve a smooth generalized trajectory from demonstrations, as well as to adapt to small variances. These probabilistic methods are not only effective but efficient in capturing characteristics embedded in demonstrated trajectories; these methods are adopted in our work to extract features in human demonstrations.

Dynamical systems are another popular approach that is robust to dynamic environment changes. Dynamic movement primitives were first developed by Ijspeert et al. [11], who designed a motor representation based on dynamical systems for encoding trajectories, and applied non-parametric regression to shape the attractor landscapes according to the demonstrated trajectories. Khansari-Zadeh and Billard [12] modulated the dynamical system using trajectories encoded by a GMM. To learn the parameters of nonlinear functions in a dynamical system, Atkeson et al. [13] proposed locally weighted regression, and Vijayakumar et al. [14] extended it to Locally Weighted Projection Regression (LWPR) to improve the learning efficiency in high dimensional spaces. Compared with GMMs, the LWPR works better for high dimensional variables and inputs that may contain irrelevant dimensions. However, approaches based on function approximation have a significant limitation, in that they cannot generalize across undemonstrated states. In this paper, we address this adaptation problem in new situations.

To automatically adapt to undemonstrated states, RL provides a natural solution by enabling agents to autonomously interact with the environment. Guenter et al. [3] combined skills learned during demonstrations with those acquired from a natural actor-critic RL module, to enable trajectory adaptability that is controlled by a dynamic system. Kober et al. [4] proposed a motor primitive approach, which initializes

the policy using human demonstrations, then self-improved policies using a Policy Learning by Weighting Exploration with the Returns algorithm. This work was extended by [15], which modulated the elementary movement through meta-parameters of its representation and proposed a RL algorithm based on a kernelized version of the reward-weighted regression to adjust robot movements to new situations. These methods simply initialize RL policy using demonstrations. But issues arise from high DoFs in robots leading to long training times. When novel states are encountered by a robot, [16], [17] acquire new demonstrations from human to correct the robot's actions according to environment changes, which is not always acceptable for long-term robot learning. As a result, how to extract good data representations only from a few demonstrations and incorporate this information into RL is the key issue for learning generalized and adaptive robot skills.

Bitzer et al. [6] proposed dimension reduction to automatically determine abstractions from demonstration data so that the learner only searches in the reduced latent space. Shon et al. [18] suggested scaled Gaussian Process Latent Variable Models (SGPLVMs) to map human and robot motion from a high-dimensional space to a low-dimensional latent variable space to find the mapping between human and robot DoFs. Cobo et al. [7] presented the automatic decomposition and abstraction from demonstration algorithm, which divides a sequential decision process into sub-tasks and extracts the most relevant features in each subtask. Different from these methods, our approach divides the entire feature space into principal and non-principal spaces; robot adaptation is then performed using non-principal features that are less restricted by the training process and contain more variations.

### III. FEATURE SPACE DECOMPOSITION AND ROBOT ADAPTATION

In this section, we discuss our FSD approach that encodes the demonstration data using principal and non-principal features. The principal features are applied to learn GMMs to reproduce initial solutions for robot adaptation based on RL. The non-principal features define a space that the robot needs to explore using RL techniques to find solutions for an unseen environment. The schematic overview of our method is illustrated in Fig. 2. Table III-A presents major notations used in this paper.

#### A. Feature Decomposition

Let  $\{x_1, x_2, \dots, x_D\}$  denote the features in the original data space  $\mathcal{O}$ . Principal Component Analysis (PCA), which has been proven to be effective for decomposing human motion data [19], is used to project  $x$  to the latent space  $\mathcal{L}$ .

To decompose principal and non-principal features,  $x$  is centered to  $\bar{x}$  by subtracting the means from each feature  $x_i, i = \{1, 2, \dots, D\}$ ; then the covariance matrix  $C$ , eigenvectors  $w_j$  and associated eigenvalues  $\lambda_j$  are calculated by:

$$\begin{aligned} C &= E(\bar{x}\bar{x}^T) \\ Cw_j &= \lambda_j w_j \end{aligned} \quad (1)$$

TABLE I  
NOTATION USED THROUGHOUT THIS WORK

Variable	Notation
$\mathcal{O}$	Original full feature space
$\mathcal{S}$	Non-principal feature space
$\mathcal{L}$	Principal/latent feature space
$x$	Features in $\mathcal{O}$
$\xi$	Features in $\mathcal{L}$
$D$	Dimensionality of $\mathcal{O}$
$n$	Dimensionality of $\mathcal{S}$
$p$	First $p$ features in the latent space $\mathcal{L}$
$K$	Number of Gaussian components
$T_1$	Threshold to decide dimensions of $\mathcal{L}$
$T_2$	Threshold to decide dimensions of $\mathcal{S}$

where  $\lambda_j$  denotes the importance of the corresponding component  $w_j, j = \{1, 2, \dots, D\}$ , and  $w_j = \{w_{j_m}\}, m = \{1, 2, \dots, D\}$  is the weight vector used to rotate the original feature set  $\{x_1, x_2, \dots, x_D\}$  into a projected space that is encoded by the new feature set  $\{\xi_1, \xi_2, \dots, \xi_D\}$ :

$$\xi_i = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_j \end{bmatrix} x_i \quad (2)$$

By accumulating the sorted eigenvalues  $\{\lambda_1, \lambda_2, \dots, \lambda_p\}$ , we get the corresponding eigenvectors  $\{w_1, w_2, \dots, w_p\}$  that contain information above a given threshold  $T_1$ , i.e.  $T_1 = 95\%$ , where  $p$  refers to the first  $p$  components covering at least 95% of the data's spread:  $\sum_{j=1}^p \lambda_j > 95\%$ .

The new feature set  $\{\xi_1, \xi_2, \dots, \xi_p\}$  contains the *principal features* in the latent space  $\mathcal{L}$  constructed by  $\{w_1, w_2, \dots, w_p\}$ . The principal features encode the main information conveyed by human demonstrations and will be used in GMM training (see next section) to obtain an initial solution for RL to adapt robot motions.

In order to extract the non-principal features, we evaluate the magnitude of contribution (MoC) of each original feature  $x$  to the principal components through calculating the length of  $x$  in the latent space. Since the eigenvalue represents the significance of each component, we rescale  $x$ 's projection in each basis and calculate the multi-dimensional length to represent the MoC ( $M$ ):

$$M_{x_i} = \sqrt{\sum_{j=1}^p (w_{j_i} \lambda_j)^2} \quad (3)$$

Then, we sort the ranking of  $M$  to extract a compact set of  $\{x_i\}$  that least contributes to the latent space. This new set of features are referred to as the *non-principal* features that convey the least relevant information of the demonstrations, and thus are less restricted by the training process and contain large variations. The non-principal features will be employed to define the action space of RL for robot adaptation.

It is noteworthy that the size of the non-principal feature set is decided by a tunable threshold  $T_2$ , so the proposed FSD approach is scalable and guaranteed to find optimal solutions

if they exist. We formulate our FSD solution as a proposition followed by a sketched mathematical proof.

*Proposition 1:* If an optimal solution  $\pi$  exists in the original full feature space  $\mathcal{O} := \{x_1, x_2, \dots, x_D\}$ , it must exist in the non-principal feature space  $\mathcal{S} := \{x_{k_1}, x_{k_2}, \dots, x_{k_n}\}$ , where  $n \leq D$  is the dimensionality determined by the FSD approach.

*Proof:* (Sketch) Sort the features in the original full space  $\mathcal{O}$  according to their MoC values to obtain a ranked original feature space  $\mathcal{O}^s = \{x_{k_1}^s, x_{k_2}^s, \dots, x_D^s\}$ . Search the non-principal feature space  $\mathcal{S} := \{x_{k_1}, x_{k_2}, \dots, x_{k_n}\}$ ,  $n \leq D$ , where  $n$  is decided by:  $\sum_{i=1}^n M_{x_{k_i}} \leq T_2$ ,  $\sum_{i=1}^{n+1} M_{x_{k_i}} > T_2$ , which is obtained by the FSD approach. If  $\pi$  exists in  $\mathcal{S}$ , then the proposition holds. If  $\pi$  doesn't exist in  $\mathcal{S}$ , then increase  $T_2$  to form a new non-principal space  $\mathcal{S}^{new}$  that includes more features ( $n^{new}$  features), and start a new search in  $\mathcal{S}^{new}$ . If no optimal solution is found, repeat the process until  $n^{new} = D$ . In this case,  $\mathcal{S}^{new} = \{x_{k_1}, x_{k_2}, \dots, x_{k_n^{new}}\} = \{x_{k_1}^s, x_{k_2}^s, \dots, x_D^s\}$ , that is  $\mathcal{S}^{new} = \mathcal{O}^s$ . If the optimal solution  $\pi$  exists in  $\mathcal{O}$ , it exists in  $\mathcal{O}^s$ , since  $\mathcal{O}^s$  is the same space with ranked features. Thus,  $\pi$  also exists in the non-principal space  $\mathcal{S}^{new}$ , and the proposition holds. ■

Intuitively, the proposition shows that the size of the non-principal feature space  $\mathcal{S}$  is adjustable, and the search space can be scaled to the entire original full feature space. As long as the threshold  $T_2$  is less than 100%,  $\mathcal{S}$  is smaller than the original  $D$ -dimensional feature space  $\mathcal{O}$ . In the worst case, all features need to be selected to guarantee that the optimal solution can be found, if it exists. Generally, the search space can be exponentially reduced by limiting the dimensions, so FSD can usually find solutions more quickly.

In the complete robot adaptation system as illustrated in Fig. 2, after the principal and non-principal feature spaces are decomposed, the principal features are applied by GMMs to learn an initial solution that can work well in unchanged environments; the non-principal features are employed by RL approaches to effectively perform robot adaptation based on the initial solution.

### B. Skill Encoding

Temporal variations are common when a human performs multiple demonstrations for the robot. We use Dynamic Time Warping (DTW) to address these time variations, which has been validated to be a robust and computationally efficient temporal alignment method [10]. In addition to the spatial information recorded during demonstration, each data point includes a time stamp. Accordingly, a projected, temporally aligned data point can be represented by  $\xi_i = \{\xi_{t,i}, \xi_{s,i}\}$ , where  $\xi_{t,i}$  is the temporal value and  $\xi_{s,i}$  is the spatial value in the principal feature space.

To extract the localized features from different segments of the task, we apply mixture modeling to encode the projected temporally aligned data, which is a widely used method for density approximation. The dataset is modeled by a mixture

of  $K$  Gaussians, with the probability density function:

$$p(\xi_i) = \sum_{k=1}^K p(k)p(\xi_i|k) \quad (4)$$

where  $p(k) = \pi_k$  is the prior, and  $p(\xi_i|k) = \mathcal{N}(\xi_i; \mu_k, \sigma_k)$  is a Gaussian component denoted by:

$$p(\xi_i|k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp\left\{-\frac{1}{2}(\xi_i - \mu_k)^T \Sigma_k^{-1} (\xi_i - \mu_k)\right\} \quad (5)$$

The *Expectation Maximization* (EM) method [20] is used to obtain the maximum likelihood estimation of parameters in the GMM. The number of Gaussians  $K$  is determined by the *Bayesian Information Criterion* (BIC) [21] approach to balance the model fitness and the training complexity.

For each GMM, the temporal components are used as queries. The spatial components (joint angles) can be derived from the mean and covariance matrix of the Gaussians:

$$\mu_k = \{\mu_{t,k}, \mu_{s,k}\}, \quad \Sigma_k = \begin{pmatrix} \Sigma_{t,k} & \Sigma_{ts,k} \\ \Sigma_{st,k} & \Sigma_{s,k} \end{pmatrix}$$

Given  $\xi_t$ , the conditional expectation  $\hat{\xi}_{s,k}$  and the estimated conditional covariance  $\hat{\Sigma}_{s,k}$  are:

$$\begin{aligned} \hat{\xi}_{s,k} &= \mu_{s,k} + \Sigma_{st,k}(\Sigma_{t,k})^{-1}(\xi_t - \mu_{t,k}) \\ \hat{\Sigma}_{s,k} &= \Sigma_{s,k} - \Sigma_{st,k}(\Sigma_{t,k})^{-1}\Sigma_{ts,k} \end{aligned} \quad (6)$$

For each Gaussian component, the responsibility for  $\xi_t$  can be computed:

$$\beta_k = \frac{p(\xi_t|k)}{\sum_{i=1}^K p(\xi_t|i)} \quad (7)$$

Combining Eqs. (6) and (7), the conditional expectation of  $\xi_s$  given  $\xi_t$  and the conditional covariance  $\Sigma_s$  are:

$$\xi_s = \sum_{k=1}^K \beta_k \hat{\xi}_{s,k}, \quad \Sigma_s = \sum_{k=1}^K \beta_k^2 \hat{\Sigma}_{s,k} \quad (8)$$

Finally, at each time step  $\xi_t$ , the space information  $\xi_s$  and the associated covariance matrix for each joint can be regressed. Note that the generalized result is not the average or part of the dataset, but rather the inherent characteristics implied in the observed movements.

### C. Reinforcement Learning

To allow robots to find solutions when the learned knowledge does not directly apply in the new environment, we use a RL module based on the non-principal features to perform autonomous robot adaptation. Since the state space and the action space in this problem are both continuous, we applied the *Gradient-Descent SARSA*( $\lambda$ ) algorithm [22] to find the satisfactory policy. The general objective of RL is to find an optimal policy that maximizes:

$$Q^\pi(s, a) = E_\pi \left\{ \sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a \right\} \quad (9)$$

where  $Q^\pi(s, a)$  is the expected return accumulated by the agent when following the policy  $\pi$  starting from the initial

state-action pair  $(s_0, a_0)$ ,  $s \in \mathcal{R}^n$  is the continuous state, and  $a \in \mathcal{R}^D$  is the continuous action.  $\gamma$  denotes a discount factor and  $r_t$  denotes the reward at time  $t$ , which is the distance between the target state  $s^*$  and the actual robot state:  $r_t = -\|s^* - s_t\|$ . For tasks in this paper, the target state is decided by the specific domain. For instance, in the writing task, the goal is to obtain a trajectory whose height is decided by the given plane.

The action-value function  $Q(s, a)$  is represented using a parameterized functional form with the parameter  $\theta_t$ :

$$Q(s, a) = \sum_{i \in F(a)} \theta(i) \quad (10)$$

where  $F(a)$  is the set of features present in  $s, a$ . When the linear tile coding method [22] is used, the gradient-descent update for action-value prediction is

$$\theta_{t+1} = \theta_t + \alpha \delta_t e_t \quad (11)$$

where

$$\begin{aligned} \delta_t &= r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \\ e_t &= \gamma \lambda e_{t-1} + \nabla_{\theta_t} Q_t(s_t, a_t) \end{aligned} \quad (12)$$

with  $e_0 = 0$  and  $\delta_t$  denotes the time step.

#### IV. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our approach, we conducted experiments using a physical humanoid robot. In this section, we first discuss the setups of our experiments. Then, we focus on the robot writing task to discuss how our approach achieves adaptation when facing new environments. In addition, experimental results on the robot drinking task are discussed to further evaluate our FSD approach. Finally, comparisons with baseline methods are presented to validate the effectiveness of the proposed method.

##### A. Experimental Setup

The robot we apply in the experiment is a NAO humanoid robot. With the robot torso and legs set to a constant position to support the robot's sitting posture, the NAO robot only uses the right arm to perform the task. In the demonstration, a human expert moves the robot arm to the desired posture with sensors recording the joint angles, and then repeats the procedure for the next gesture. For each experiment, the task is demonstrated four times. Each demonstration lasts around 10 seconds, generating 100~200 data points. We adopt the *Forward Kinematic Model* of the NAO robot [23] to compute the end-effector positions in the function of the joint angles.

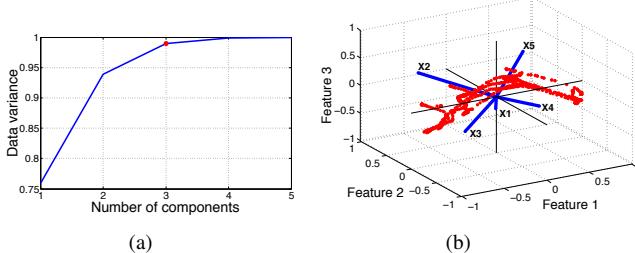


Fig. 3. (a) Estimation of the number of principal components. (b) Data and original features in the latent space represented by three principal features.

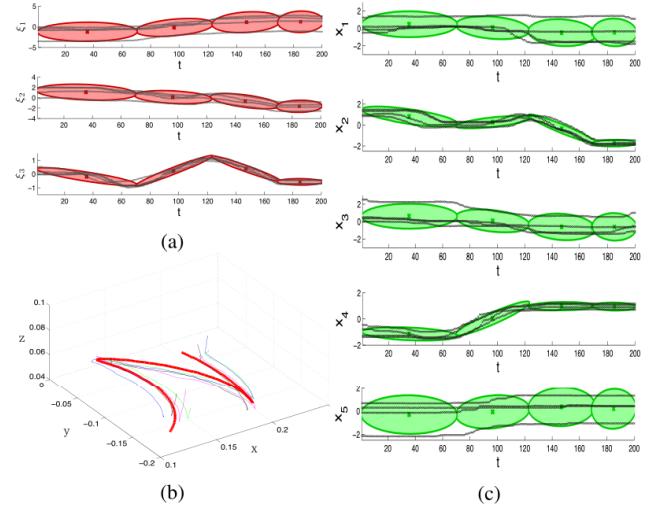


Fig. 4. (a) GMM encoding in the principal/latent space. (b) Generalized writing trajectory (red line) and four demonstrated trajectories (thin lines). (c) GMMs reprojected onto the original feature space.

#### B. Experiments on Robot Writing Tasks

1) *Feature Space Decomposition*: When movements are demonstrated to the NAO robot, we separate the features (i.e., joint angles) encoded in the observed data into principal and non-principal features using the proposed FSD method. Fig. 3(a) shows that if we want to encode 95% of the information in the original feature space (i.e.,  $T_1 = 95\%$ ), the top three principal components are sufficient. We quantitatively inspect the features that contribute the least to the latent feature space by analyzing their MoC values, as shown in Table II. After sorting the MoC values of the original features, their ranking in ascending order is  $\{M_{x_1}, M_{x_5}, M_{x_2}, M_{x_3}, M_{x_4}\}$ . The threshold  $T_2$  is empirically set to 25% to filter the least important features. It is observed that  $\mathcal{S} := \{x_1, x_5\}$  satisfies  $M_{x_1} + M_{x_5} < T_2$  and  $M_{x_1} + M_{x_5} + M_{x_2} > T_2$ , so that  $x_1$  and  $x_5$  are selected to represent the non-principal feature space. These features are applied to construct the new action space for RL to adapt robot motions in new environments.

TABLE II  
MoC VALUES OF THE ORIGINAL FEATURES.

MoC (M)	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Writing task	0.86%	22.10%	23.01%	35.97%	18.05%
Drinking task	27.53%	6.20%	23.16%	22.09%	21.02%

For each feature  $x$ , if it has  $k$  DoFs, i.e.  $k$  sets of angles, then the dimension of the action space is  $k^n$ , where  $n$  is the number of features in the action space. In this experiment, the original feature space  $\mathcal{O} := \{x_1, x_2, x_3, x_4, x_5\}$  defines a  $k^5$ -dimensional action space given that the NAO's arm has 5 joints, while the non-principal features  $\mathcal{S} := \{x_1, x_5\}$  obtained by our FSD approach constructs an action space with only  $k^3$  dimensions. As a result, our proposed method reduces the action states exponentially by reducing  $n$ , which accordingly enables the FSD-based RL approaches to find optimal solutions more quickly in most cases. Meanwhile,

the FSD approach is effective because it can be scaled to the entire original feature space and is thus guaranteed to find an optimal solution if it exists in the original feature space.

2) *Generalization and Robot Adaptation*: After the principal features are extracted, we encode them using GMMs to obtain the regression results. By reprojecting the regression results back onto the original space, the joint movements are generated, as illustrated in Fig. 4 where the variability and expected trajectory are represented by the ellipses and centers of GMMs, respectively. Features with low variability across demonstrations can be interpreted as more relevant to the task than features with high variability, which is consistent with Fig. 3(b) that  $x_2, x_3, x_4$  are more relevant than  $x_1$  and  $x_5$ . In addition, the jerky and noisy data in demonstrations (short lines above the writing plane) is smoothed as shown by the generalized trajectory in Fig. 4.

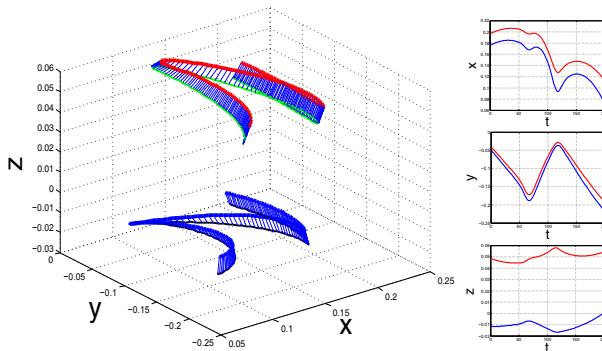


Fig. 5. A new trajectory (blue) is learned by RL to adapt to the new plane ( $height = 0.025m$ ); the corresponding marker trajectory (black) is also shown. Generalized hand trajectory (red) and the marker trajectory (green) in the standard situation ( $height = 0.04m$ ) are presented for comparison. The small blue arrows between the hand trajectories and the marker trajectories represent the marker's orientation. Time series in each dimension are shown in the second row.

The generalized trajectory captures the important features of the writing task, so we input it into RL as an initial policy. For the writing task, the parameters of the Gradient Descent SARSA( $\lambda$ ) are given, with the probability of random action  $\epsilon = 0.1$ , the step size  $\alpha = 0.5$ , the trace decay parameter  $\lambda = 0.9$ , and the discount rate  $\gamma = 0.9$ . After learning for less than 5000 episodes, the robot successfully finds the policy to adapt to the new writing plane which is lower than the plane during demonstrations, as illustrated in Fig. 5. By controlling the continuous velocity of each joint, the angle of robot joints at each time are decided. Results show that only the height of the hand ( $z$  direction) is changed to adapt to the new plane, while the trajectories in the  $x-y$  plane are preserved. This validates that the proposed FSD approach is able to capture the important features of the task, and enables the robot to adapt to different planes.

We perform additional experiments to further validate the effectiveness of our FSD method by presenting the NAO robot with new writing planes, as shown in Fig. 6. The two trajectories ( $0m/0^\circ, 0.02m/25^\circ$ ) in Fig. 6(a) illustrate that even though we train the robot in just one plane parallel to the demonstration plane, the robot can accomplish the task

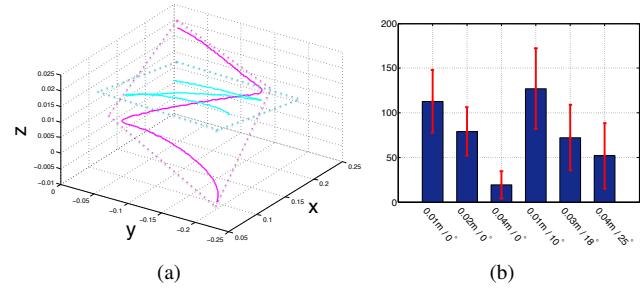


Fig. 6. (a) Testing results of writing task in the parallel and tilted planes. (b) Average number and standard deviation of trials needed to find solutions for 6 different testing planes ( $x$  axis: plane height/angle).

in different unseen situations such as tilted planes. When the robot successfully writes in the desired plane without failure (colliding with the plane that prevents smooth writing, being trapped in a small area preventing completion of the writing trajectory, or losing touch with the plane), we consider the solution is found so that the robot is able to accomplish the task. The number of trials to find the solution is illustrated in Fig. 6(b). We provide several exemplary snapshots in Fig. 7 to illustrate that the NAO robot is able to adapt to unseen changes of the environment. This experiment indicates that by learning from outside the restricted latent space, the FSD approach implements our new perspective that enables more effective autonomous robot adaptation following the PbD and RL framework.

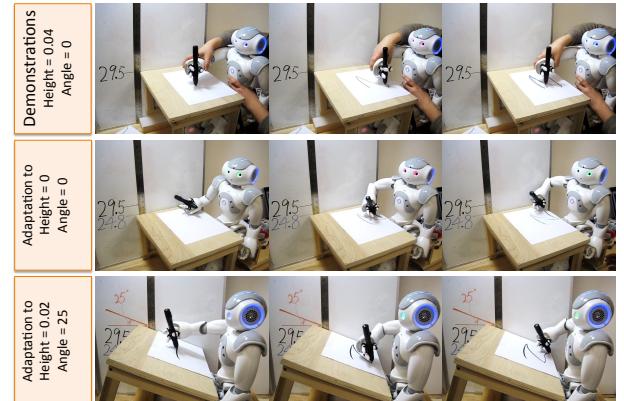


Fig. 7. Human demonstrations and experimental results of robot adaptation in two different new environments using lifted and tilted tables in the robot writing tasks.

### C. Experiments on Robot Drinking Tasks

1) *Feature Space Decomposition*: We further validate our FSD approach's performance on the robot drinking task. In this experiment, a human instructor demonstrates a drinking task four times by moving the robot's right arm, as shown in Fig. 9. The trajectory passes through three points: the start point, the destination point (the robot's mouth), and the end point, as highlighted in Fig. 8(a). We decompose the original feature space using our FSD approach. Then, the principal features are encoded using GMMs to generate a generalized

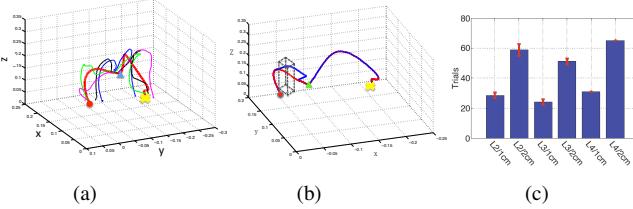


Fig. 8. (a) The generalized drinking trajectory is shown by the red line, while the thin lines represent the demonstrated trajectories. The generalized trajectory starts from an initial point represented by the red dot, passes the robot's mouth at the green triangle, and ends at the yellow cross. (b) Testing results of the drinking task with a new obstacle in the environment. The obstacle is placed on the path of the generalized trajectory (red); the adapted trajectory (blue) avoids the obstacle by passing to the side along with moving toward the target. (c) Average number and standard deviation of trials needed to converge to solutions for 2 different obstacles in 3 locations ( $x$  axis represents location/obstacle size).

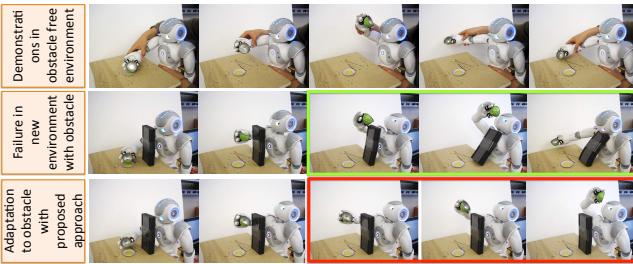


Fig. 9. Human demonstrations and experimental results of robot adaptation with and without applying RL in unseen environments in the robot drinking tasks. (Head motion is pre-computed.)

trajectory. The features are ranked as  $\{x_2, x_5, x_4, x_3, x_1\}$ , as shown in Table II, according to their MoC values. Similar to the robot writing task, the threshold  $T_2 = 25\%$  is used to filter the least important features. Since feature  $x_2$  contributes less than 20%, and  $M_{x_2} < T_2$  and  $M_{x_2} + M_{x_5} > T_2$  are satisfied, the non-principal feature space  $\mathcal{S} := \{x_2\}$  is selected by our FDS approach. Accordingly, the dimension of the RL action space is exponentially reduced from  $k^5$  to  $k$ , given that each feature has  $k$  DoFs. Since the number of non-principal features in space  $\mathcal{S}$  depends on  $T_2$ , by increasing  $T_2$ , the size of  $\mathcal{S}$  can be enlarged. Thus, the FSD approach guarantees that the RL methods can find optimal solutions. In addition, because the search of optimal solutions typically starts with a small non-principal feature space, our approach can usually find the solutions more quickly.

*2) Generalization and Robot Adaptation:* To evaluate the performance of the entire robot adaptation system, we equally divide the whole trajectory into 200 segments, and assign four different locations:  $L1$ : segment 40-50,  $L2$ : segment 50-60,  $L3$ : segment 60-90,  $L4$ : 90-100. In the RL training, a 2cm width obstacle is placed in  $L1$  along the trajectory. The robot takes around 5000 episodes to successfully learn the policy of avoiding the obstacle. To validate the learned solution in different situations, we move the obstacle to  $L2 \sim L4$ , i.e., the black box ( $L2$ ) as depicted in Fig. 8(b). The generated trajectory (the blue line in 8(b)) shows that the robot adapts to the environment and is able

to avoid the obstacle smoothly. Fig. 8(c) illustrates the number of trials to find a satisfactory solution for six different testing situations, using the same RL parameters as in the writing experiment.

In Fig. 9, experimental results before and after applying RL methods are intuitively compared in the second and third rows. Results without using RL show that the robot runs into the obstacle placed along the GMM generalized trajectory, as shown by the images in the green rectangle. Results after applying RL, as shown by the images in the red rectangle in Fig. 9, demonstrate that the robot successfully bypasses the obstacle by its side and finishes the task by bringing the cup to its mouth.

#### D. Comparison with Baseline Methods

We conducted additional empirical studies to compare our FSD approach with two baseline robot adaptation methods:

- **Adaptation method based on the original feature space:** This baseline method applied in the experiment is based on the original feature space  $\mathcal{O} = \{x_1, x_2, \dots, x_D\}$  [3]. The dimensionality of this space is  $k^D$  given that each feature has  $k$  DoFs.
- **Adaptation method using the principal feature space:** This baseline approach uses the principal features (i.e., in the latent space) to represent the demonstration data [6], which is the set  $\mathcal{L} = \mathcal{O} - \mathcal{S}$  with  $k^{D-n}$  dimensions.

It is observed from Fig. 10 that the baseline method base on the original full feature space does not converge within  $10^4$  trials, while the other two approaches based on the non-principal and principal feature spaces converge, meaning that the reward value does not change much with the increase of the number of trials. This shows that methods based on the original feature space are not efficient, and usually cannot find a solution within a reasonable period of time, as in our robot writing and drinking tasks. A plausible explanation is that this baseline method suffers from the curse of dimensionality, which demonstrates the need of an approach (e.g., our FSD algorithm) that can allow robots to search solutions in a low-dimensional space for adaptation.

Fig. 10 also shows that although both our approach based on the non-principal features and the baseline method using the principal features (i.e., in the latent space) converge, the converged accumulated reward of our method is greater than that of the baseline method (e.g., red lines versus blue lines in Fig. 10). This phenomenon demonstrates that our FSD-based adaptation approach is more effective than the conventional methods based on the principal/latent space, since it is able to find solutions with better rewards. In some tasks (e.g., in our robot drinking task), the baseline approach based on the conventional latent feature space may not be able to find an optimal solution (although it can converge), as shown by the converged smaller reward value in Fig. 10. On the other hand, if an optimal solution exists in the original feature space, our FSD-based approach is able to find it due to its scalability to the full feature space.

In addition, comparing to the baseline approach using the principal/latent feature space in Fig. 10, we can observe that

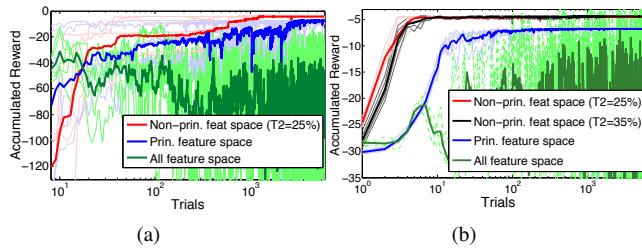


Fig. 10. Accumulated rewards versus the number of trials in (a) the robot writing task and (b) the robot drinking task. Red and black lines: our FSD approach; Blue line: baseline method based on the principal feature space; Green line: baseline method using all features.

the proposed FSD-based algorithm achieves better efficiency by converging to an optimal solution more quickly in both robot writing and drinking tasks (while the baseline method using all original features fails to converge within  $10^4$  trials). A plausible explanation is that the principal feature space is optimized and thus restricted by the training environments, while the non-principal feature space usually contains more variations and can have a better chance of including solutions for new situations that are different from the environments during the training process. It is also observed that in Fig. 10(b), the larger  $T_2$  (35%) results in the longer convergence time, because a larger search space is created for RL training, which suggests that selecting a small threshold yields better learning performance.

## V. CONCLUSION AND DISCUSSION

We introduce a novel Feature Space Decomposition (FSD) approach, within the PbD and RL learning methodology, to enable robots to effectively adapt to environment changes. By computing the magnitude of contribution of each feature, our FSD methods divides the original feature space into non-principal and principal feature spaces. The principal features are encoded by GMMs to generalize a solution as the initial policy of RL. Then, we construct the RL action space based on the non-principal features obtained by our FSD approach, which guarantees that an optimal solution (if it exists) can be found for robot adaptation to new environments. Experiments using physical robots are performed on the robot writing and drinking tasks. Results have validated that our FSD approach enables more effective autonomous robot adaptation than the conventional techniques based on the latent feature space. In addition, the FSD-based robot adaptation approach can often find solutions more quickly.

In future work, we plan to incorporate information of the training environments relevant to human demonstrations into our FSD-based robot adaptation method. Although this information has the potential to provide valuable cues to estimate the environment changes, it may also aggravate the curse of dimensionality issue. Thus, more sophisticated approaches need to be developed to tackle this interesting but more challenging problem. Another promising future research is to validate the applicability of our FSD-based robot adaptation approach in more complicated tasks (e.g., daily assistance such as cooking) using more powerful humanoid robots (e.g.

Baxter and Meka robots).

## REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Handbook of Robotics*, ch. 59: *Robot Programming by Demonstration*. Springer, Jan 2008.
- [2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, May 2009.
- [3] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *RSJ Advanced Robotics*, pp. 1521–1544, 2007.
- [4] J. Kober and J. Peters, "Imitation and reinforcement learning: Practical learning algorithms for motor primitives in robotics," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 55–62, 2010.
- [5] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, pp. 1238–1274, July 2013.
- [6] S. Bitzer, M. Howard, and S. Vijayakumar, "Using dimensionality reduction to exploit constraints in reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.
- [7] L. C. Cobo, K. Subramanian, C. L. Isbell Jr., and A. L. Thomaz, "Abstraction from demonstration for efficient reinforcement learning in high-dimensional domains," *Artificial Intelligence*, vol. 216, pp. 103–128, 2014.
- [8] T. Inamura, H. Tanie, and Y. Nakamura, "Keyframe compression and decompression for time series data based on the continuous Hidden Markov Model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- [9] A. Vakanski, I. Manteghi, A. Irish, and F. Janabi-Sharifi, "Trajectory learning for robot programming by demonstration using Hidden Markov Model and dynamic time warping," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, pp. 1039–1052, Aug 2012.
- [10] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 2, pp. 286–298, 2007.
- [11] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation*, pp. 1398–1403, 2002.
- [12] S. M. Khansari-Zadeh and A. Billard, "Learning stable non-linear dynamical systems with Gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [13] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning for control," *Artificial Intelligence Review*, vol. 11, pp. 75–113, 1997.
- [14] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural Computation*, vol. 17, no. 12, pp. 2602–2634, 2005.
- [15] J. Kober, E. Oztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *International Joint Conference on Artificial Intelligence*, 2011.
- [16] D. H. Grollman and O. C. Jenkins, "Dogged learning for robots," in *IEEE International Conference on Robotics and Automation*, 2007.
- [17] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2007.
- [18] A. P. Shon, K. Grochow, and R. P. Rao, "Robotic imitation from human motion capture using Gaussian processes," in *IEEE RAS International Conference on Humanoid Robots*, 2005.
- [19] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM," in *International Conference on Machine Learning*, pp. 105–112, 2005.
- [20] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc, 2006.
- [21] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 239–472, 1978.
- [22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [23] N. Kofinas, "Complete analytical inverse kinematics for NAO," in *International Conference on Autonomous Robot Systems and Competitions*, 2013.