Equivariant Motion Manifold Primitives

Anonymous Author(s)
Affiliation
Address
email

Abstract: Existing movement primitive models for the most part focus on representing and generating a single trajectory for a given task, limiting their adaptability to situations in which unforeseen obstacles or new constraints may arise. In this work we propose Motion Manifold Primitives (MMP), a movement primitive paradigm that encodes and generates, for a given task, a continuous manifold of trajectories each of which can achieve the given task. To address the challenge of learning each motion manifold from a limited amount of data, we exploit inherent symmetries in the robot task by constructing motion manifold primitives that are equivariant with respect to given symmetry groups. Under the assumption that each of the MMPs can be smoothly deformed into each other, an autoencoder framework is developed to encode the MMPs and also generate solution trajectories. Experiments involving synthetic and real-robot examples demonstrate that our method outperforms existing manifold primitive methods, in some cases by significant margins.

Keywords: Movement primitives, Manifold, LfD, Equivariance

1 Introduction

Learning basic motion skills as movement primitives has been an enduring focus of learning from demonstration (LfD) research [1, 2, 3], particularly the problem of constructing movement primitive models that are adaptable to a wide range of situations such as when unforeseen obstacles or new constraints arise. Existing approaches to movement primitives include dynamic movement primitives [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], stable dynamical systems [14, 15, 16, 17, 18, 19, 20, 21, 22], methods based on Gaussian processes [23, 24, 25] and Gaussian mixture models [26, 27, 28, 29], and other methods [30, 31, 32].

The limited adaptability of existing primitive models can be attributed in large part to the fact that they are designed to encode and generate a single trajectory for a given task. Such single trajectory-based primitives fundamentally have limited adaptability, because there is no alternative if that trajectory becomes infeasible in the new environment (e.g., if an unforeseen obstacle blocks that trajectory). While, for dynamical system-based methods, it is possible to incorporate, e.g., obstacle avoidance potential function terms into the dynamics model [33, 34, 35, 36], the resulting motions may violate other task constraints. To obtain highly adaptable motion primitives, what is needed is a method that can encode a set of multiple trajectories that can perform the given task.

In this paper, we propose to learn a continuous manifold of motion trajectories that can perform the given task, which we refer to as Motion Manifold Primitives (MMP). Since multiple successful trajectories are encoded in MMP, even when certain trajectories are obstructed by obstacles or violate constraints, the other viable trajectories can be found in MMP. As such, the MMP is highly adaptable, although more diverse demonstration data are needed for training, compared to the single trajectory-based primitives.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

Given a set of task-trajectory paired data – where multiple demonstration trajectories are collected for a single task parameter τ –, our objective is to learn a set of manifold primitives $\{\mathcal{M}_\tau\}$ for all τ. However, given a limited amount of demonstration data, learning accurate manifolds and their boundaries is a very challenging task. In fact, the TC-VAE [37] is the first work to adopt this manifold primitives approach but shows less-than-desirable performance given a small dataset as we show later in our experiments.

In this paper, we develop a data-efficient motion manifold primitives learning algorithm, where we adopt the autoencoder-based manifold learning framework [38, 39, 40, 41]. First, we assume $\mathcal{M}_\tau$ for all τ are homeomorphic (i.e., can be smoothly deformed into each other), so that a shared latent coordinate space $\mathcal{Z}$ can be found. In practice, we assume that the latent coordinate variable $z \in \mathcal{Z}$ is independent of τ, so that $p(z|\tau) = p(z)$ for all τ, and the support of $p(z)$ becomes the shared latent coordinate space $\mathcal{Z}$. To enforce this condition, we propose an *independence regularization term* in autoencoder training.

Second, we propose *Equivariant Motion Manifold Primitives (EMMP)*, considering inherent symmetry in the robot task. For example, consider a water-pouring task where the initial cup and bottle's position is defined as the task parameter. Given a symmetry transformation on τ that preserves the relative positional relation between the cup and bottle (e.g., rotating bottle around the cup), the set of water-pouring trajectories, the motion manifold primitives $\mathcal{M}_\tau$, should also be transformed in a consistent manner, and more precisely, *equivariantly*. We show that using an invariant encoder and equivariant decoder can guarantee the equivariance of the MMP and propose a strategy to construct invariant and equivariant mappings.

Through extensive experiments involving synthetic and real-world robot water-pouring experiments, we compare our EMMP model with the existing manifold primitives method TC-VAE [37] through a systematic evaluation on (i) manifold learning, (ii) independence between z and τ, (iii) density learning, and (iv) success rate. We confirm that our method outperforms TC-VAE by a significant margin, and via a thorough ablation study, we show that using an invariant encoder and equivariant decoder is the most critical factor in performance improvement.

2 MMPs: Motion Manifold Primitives

In this section, given a configuration space $\mathcal{Q}$, we consider a motion as a discrete and fixed-length configuration trajectory $x = \{q_t \in \mathcal{Q}\}_{t=1}^T$; the space of all trajectories is denoted by $\mathcal{X} := \mathcal{Q}^T$. Let a compact space $\mathcal{T}$ represent a space of parameters that define specific tasks. We assume that, for each $\tau \in \mathcal{T}$, a set of motions that can perform the given task forms an m-dimensional differentiable manifold, which we refer to as the Motion Manifold Primitives (MMP) denoted by $\mathcal{M}_\tau$, and that is a sub-manifold lying in the ambient space $\mathcal{X}$. The motion manifold $\mathcal{M}_\tau$ is in general assumed to have multiple connected components with their boundaries. We assume that we are given a set of task-trajectory paired data, denoted by $\mathcal{D} := \{(\tau_i, x_{ij}) \in \mathcal{T} \times \mathcal{X}\}_{i=1,\dots,N, j=1,\dots,M_i}$, where M_i represents the total number of trajectories for the task parameter τ_i and x_{ij} is sampled from $\mathcal{M}_{\tau_i}$.

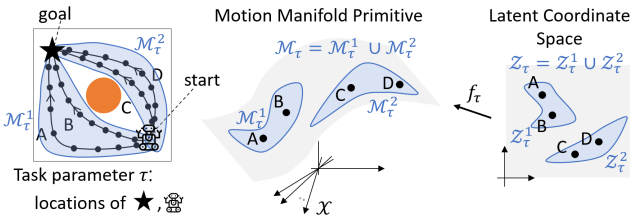

Figure 1: Trajectories that achieve a task τ are assumed to form a manifold $\mathcal{M}_\tau$. $\mathcal{M}_\tau$ consists of two disjoint component $\mathcal{M}_\tau^1$ and $\mathcal{M}_\tau^2$.

2.1 MMP described via Manifold and Density

This section introduces how to represent the set of motion manifold primitives $\mathcal{M}_\tau \subset \mathcal{X}$ for all $\tau \in \mathcal{T}$. We use the following two components: (i) a differentiable mapping f to represent a set

of coordinate systems for the manifolds and (ii) a conditional probability density function p that specifies their boundaries.

First, consider a differentiable mapping $f : \mathbb{R}^m \times \mathcal{T} \to \mathcal{X}$ where $f_\tau : \mathbb{R}^m \to \mathcal{M}_\tau \subset \mathcal{X}$ defined as $f_\tau(z) := f(z, \tau)$ is a coordinate system for the sub-manifold $\mathcal{M}_\tau$. Second, consider a conditional probability density function defined in $\mathbb{R}^m$ for each τ, denoted by $p(z|\tau)$. We assume the support of $p(z|\tau)$ is a compact subspace of $\mathbb{R}^m$ denoted by $\mathcal{Z}_\tau$. Then each motion manifold for τ can be represented as $\mathcal{M}_\tau = f(\mathcal{Z}_\tau, \tau)$; the motion manifold primitive $\mathcal{M}_\tau$ is said to be parametrized by the mapping f and the density $p(z|\tau)$ or parametrized by f and $\mathcal{Z}_\tau$ (see Figure 1).

2.2 Autoencoder-based Manifold and Density Learning

We adopt the autoencoder framework for learning $\mathcal{M}_\tau$ via $f(z, \tau)$ and $p(z|\tau)$, where f is considered as the decoder and p as the latent space density in $\mathbb{R}^m$. An additional component that specifies the coordinates of an input trajectory $x \in \mathcal{M}_\tau$ needs to be introduced, called an encoder, and we denote it by $g : \mathcal{X} \times \mathcal{T} \to \mathbb{R}^m$ such that $z = g(x, \tau)$. We use parametric models (e.g., neural networks) for encoder g_ϕ, decoder f_θ, and density $p_\gamma(z|\tau)$, where θ, ϕ, and γ represent the model parameters. We propose a two-step approach where we first learn the coordinate systems, i.e., f_θ, g_ϕ, and then learn the density, i.e., $p_\gamma(z|\tau)$.

The standard autoencoder reconstruction loss can be employed to learn f_θ, g_ϕ, that is an expectation of $d^2_{\mathcal{X}}(f_\theta(g_\phi(x_{ij}, \tau_i), \tau_i), x_{ij})$ where $d_{\mathcal{X}}(\cdot, \cdot)$ is a proper distance measure on $\mathcal{X}$. Minimizing the reconstruction loss results in that $f_\theta(g_\phi(\mathcal{M}_\tau, \tau), \tau) \approx \mathcal{M}_\tau$ for the ground truth manifold $\mathcal{M}_\tau$. $f_\theta(\cdot, \tau)$ becomes a coordinate system for $\mathcal{M}_\tau$ and $g_\phi(\mathcal{M}_\tau, \tau)$ becomes the coordinate space $\mathcal{Z}_\tau$. Secondly, we can learn $p_\gamma(z|\tau)$ given a trained encoder g_ϕ via standard likelihood maximization framework, i.e., maximizing an expectation of $\log p_\gamma(g_\phi(x_{ij}, \tau_i)|\tau_i)$.

2.3 Homeomorphic Manifold Assumption

Learning densities for all τ, i.e., $p(z|\tau)$ is challenging given the small training dataset $\mathcal{D}$. In this section, we introduce a homeomorphic manifold assumption to make the density learning problem more tractable. We assume that $\mathcal{M}_\tau$ for each τ are homeomorphic to each other and there exists a latent coordinate variable z statistically independent of τ, i.e. $p(z) = p(z|\tau)$ for all τ, which results in a shared latent coordinate space $\mathcal{Z} = \mathcal{Z}_\tau$ for all τ. The training of autoencoder and density can be greatly simplified under this assumption.

First, we can restrict the encoder's input space to $\mathcal{X}$, i.e., $g : \mathcal{X} \to \mathbb{R}^m$, because τ should not contribute to z. Second, $p(z|\tau)$ can be replaced by a shared model $p(z)$. Using $g(x)$ and $p(z)$, we may sequentially train the autoencoder and density, however, the reconstruction loss alone does not guarantee the statistical independence between z and τ, hence learning the density $p(z)$ can fail.

In this section, we introduce an independence regularization term for autoencoder training, so that z and τ become independent and $\mathcal{Z} = \mathcal{Z}_\tau$ for all τ:

$$\mathcal{R}(\theta, \phi) := \frac{\mathbb{E}_{(\cdot, x_{ij}) \in \mathcal{D}} \left[\| g_\phi(x_{ij}) - g_\phi(f_\theta(g_\phi(x_{ij}), \tau)) \|^2 \right]}{\mathbb{E}_{(\cdot, x_{ij}) \in \mathcal{D}} \| g_\phi(x_{ij}) \|^2}, \tag{1}$$

where τ is randomly sampled from the uniform distribution on $\mathcal{T}$. This loss enforces that $g(f(z, \tau))$ does not depend on τ, and eventually, τ does not contribute to z, and becomes independent of z. The denominator makes the regularization term invariant to the latent value scale. This regularization term $\mathcal{R}$ is added to the reconstruction loss with a proper regularization coefficient.

3 EMMPs: Equivariant Motion Manifold Primitives

In this section, we assume that, when symmetry transformation is applied to the task parameter τ, the motion manifold primitive $\mathcal{M}_\tau$ should also be transformed equivariantly. We denote a symmetry group by H where the group operation between two elements $h_1, h_2 \in H$ is written as $h_1 h_2 \in H$.

We assume that the symmetry transformation on $\mathcal{T}$ is defined as the group action $\phi : H \times \mathcal{T} \to \mathcal{T}$, which is simply denoted by $\phi(h, \tau) = h \cdot \tau$. Similarly, the symmetry transformation on $\mathcal{X} \times \mathcal{T}$ is denoted by $(x, \tau) \mapsto h \cdot (x, \tau)$. Let $\left[\,\cdot\,\right]_x$ denote the x component, i.e., $\left[(x, \tau)\right]_x = x$. The ground truth motion manifold primitive is denoted by $\mathcal{M}_\tau$, while the learned motion manifold primitive is denoted by $\widehat{\mathcal{M}}_\tau$ parametrized by a decoder $f(z, \tau)$ and $p(z|\tau)$.

3.1 Invariant Encoder and Equivariant Decoder

We begin this section with the definition of the equivariant motion manifold primitive:

Definition 3.1. Suppose $\mathcal{M}_\tau$ is a motion manifold primitive. Given a transformed task parameter $\tau \mapsto h \cdot \tau$, if the motion manifold $\mathcal{M}_\tau$ is equivariantly transformed, i.e., $\mathcal{M}_{h\cdot\tau} = \{\left[h \cdot (x, \tau)\right]_x \mid x \in \mathcal{M}_\tau\}$, then the primitive is called an Equivalent Motion Manifold Primitive (EMMP); see Figure 2.

We show that using an invariant encoder and an equivariant decoder can guarantee the equivariance of the resulting MMP; we first provide their definitions:

Definition 3.2. An encoder $g : \mathcal{X} \times \mathcal{T} \to \mathbb{R}^m$ is invariant if $g(h \cdot (x, \tau)) = g(x, \tau)$ for all $h \in H$ and $x \in \mathcal{X}$.

Definition 3.3. A decoder $f : \mathbb{R}^m \times \mathcal{T} \to \mathcal{X}$ is equivaraint if $f(z, h \cdot \tau) = \left[h \cdot (f(z, \tau), \tau)\right]_x$ for all $h \in H$ and $\tau \in \mathcal{T}$.

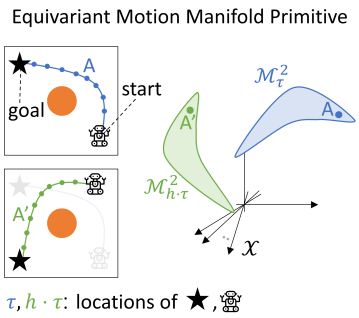

Figure 2: Given a symmetry transformation to the task parameter $\tau \mapsto h \cdot \tau$ where the relative distances between the robot, goal, and obstacle are preserved, the MMP is equivariantly transformed.

Denoting the ground truth motion manifold by $\mathcal{M}_\tau$, the latent coordinate space $\mathcal{Z}_\tau$ defined as the support of $p(z|\tau)$ can be specified by an encoder $g(x, \tau)$ as follows: $\mathcal{Z}_\tau = (\bigcup_{x \in \mathcal{M}_\tau} g(x, \tau))$ or $g(\mathcal{M}_\tau, \tau)$. If $\mathcal{M}_\tau$ is equivariant, then an invariant encoder produces invariant coordinate space:

Proposition 3.1. *Suppose $\mathcal{M}_\tau$ is an EMMP. If g is invariant, then $\mathcal{Z}_\tau = \mathcal{Z}_{h\cdot\tau}$ for all $h \in H$.*

In addition to the encoder invariance condition, the learned MMP parametrized by a decoder $f(z, \tau)$ and $\mathcal{Z}_\tau$, denoted by $\widehat{\mathcal{M}}_\tau$, is equivariant if a decoder f is equivariant:

Proposition 3.2. *Suppose $\mathcal{M}_\tau$ is an EMMP. If g is invariant and f is equivariant, then the MMP parametrized by f and $\mathcal{Z}_\tau$, i.e., $\widehat{\mathcal{M}}_\tau = f(\mathcal{Z}_\tau, \tau)$ where $\mathcal{Z}_\tau = g(\mathcal{M}_\tau, \tau)$, is equivariant.*

Therefore, by constructing an invariant encoder and equivariant decoder, we can guarantee the equivariance of the learned motion manifold primitives. If the ground truth MMP is equivariant, one can naturally expect that this equivariance guarantee will increase the manifold learning accuracy.

3.2 Construction of Invariant and Equivariant Mappings

In this section, we propose a method for converting arbitrary encoder and decoder models to ones that are invariant and equivariant to the symmetry group H. Let $G_\phi : \mathcal{X} \times \mathcal{T} \to \mathbb{R}^m$ and $F_\theta : \mathbb{R}^m \times \mathcal{T} \to \mathcal{X}$ be arbitrary parametric models for encoder and decoder, respectively. Assuming we can find an equivariant module $\bar{h} : \mathcal{T} \to H$ such that $\bar{h}(h \cdot \tau) = h\bar{h}(\tau)$ for all $h \in H, \tau \in \mathcal{T}$, an invariant encoder and equivariant decoder can be constructed as follows:

Proposition 3.3. *An encoder $g : \mathcal{X} \times \mathcal{T} \to \mathbb{R}^m$ defined as $g_\phi(x, \tau) := G_\phi(\bar{h}(\tau)^{-1} \cdot (x, \tau))$ is invariant and a decoder $f : \mathbb{R}^m \times \mathcal{T} \to \mathcal{X}$ defined as $f_\theta(z, \tau) := \left[\bar{h}(\tau) \cdot (F_\theta(z, \bar{h}(\tau)^{-1} \cdot \tau), \bar{h}(\tau)^{-1} \cdot \tau)\right]_x$ is equivariant.*

When we adopt the homeomorphic manifold assumption and use a parametric model $G_\phi : \mathcal{X} \to \mathbb{R}^m$, the invariant encoder can be defined as follows: $g_\phi(x, \tau) := G_\phi(\left[\bar{h}(\tau)^{-1} \cdot (x, \tau)\right]_x)$, where g_ϕ needs to take τ as an input while G_ϕ does not. How to construct the module $\bar{h}$ is problem-specific, and we provide examples in the subsequent experiment section.

4 Experiments

In this section, we compare our EMMP framework mainly with the existing manifold primitives method, TC-VAE [37], via both synthetic and real-world robot experiments. Additionally, we compare MMP, MMP + indep, EMMP, and EMMP + indep, where the EMMP uses an invariant encoder and equivariant decoder, and '+ indep' indicates adding the independence regularization term in autoencoder training. When training TC-VAE, MMP, and MMP + indep, we always apply random data augmentation on τ. Throughout, we use Gaussian Mixture Model (GMM) to fit $p(z)$.

Evaluation metrics: We report four evaluation metrics. First, the manifold learning accuracy is measured by the Reconstruction Error (RE). Second, to measure the degree of independence, we report estimated Mutual Information (MI) between z and τ, by using MINE [42]. Third, we compute the probability density learning performance by the Negative Log-Likelihood (NLL) measured in the data space. Lastly, we report the task Success Rate (SR), of which criteria is task-dependent. Details on the computations of these measures are described in Appendix C.1.

4.1 Goal-Reaching Task of a Planar Mobile Robot

In this section, we consider a goal-reaching task for a planar mobile robot, given a cross-shaped wall, without colliding with the wall; the robot is required to enter through one of the two closest passages (see Figure 3). As shown, it is reasonable to assume a set of feasible trajectories form a continuous manifold. Our goal is to learn the motion manifold primitive for an arbitrarily rotated wall and robot's initial position.

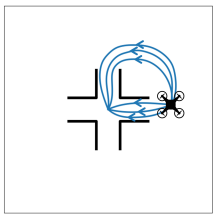

Figure 3: Demo trajectories.

The configuration space is $\mathcal{Q} = \mathbb{R}^2$, the trajectory space is $\mathcal{X} = \mathbb{R}^{2T}$. The task parameter τ consists of the initial position of the mobile robot, denoted by (q_r^1, q_r^2), and the rotated angle of the wall axis $\hat{x}_{\mathrm{w}}$ with respect to $\hat{x}_s$, denoted by ω_{w} (see Figure 4 *Left*). To make demonstration data, a human demonstrator has drawn feasible trajectories, 6 trajectories for 75 randomly given task parameters. Figure 3 illustrates example demonstration trajectories. More details on data generation and data split for training, validation, and testing are included in Appendix C.2.2.

Symmetries: There exist three symmetries that preserve the relative geometric relation between the wall and initial mobile robot position: (i) flipping of the robot over the wall axis $\hat{x}$, (ii) rotation of the robot around the origin by 90, 180, and 270 degrees, and (iii) rotation of the wall and mobile robot around the origin by the same amount (see Figure 4 *Right*). These symmetry transformations can be described as group actions with the symmetry group $H := p4m \times \mathrm{SO}(2)$, where $p4m$ is a specific type of wallpaper group and SO(2) is the group of 2×2 rotation matrices. We denote $p4m := \{(i,j) | i \in \{0,1\}, j \in \{0,1,2,3\}\}$, where $i \in \{0,1\}$ represents flipping and $j \in \{0,1,2,3\}$ represents $n\pi/2$ rotation. Details on the group operations in H and group actions on $\mathcal{T}$ and $\mathcal{X} \times \mathcal{T}$ are discussed in Appendix C.2.1.

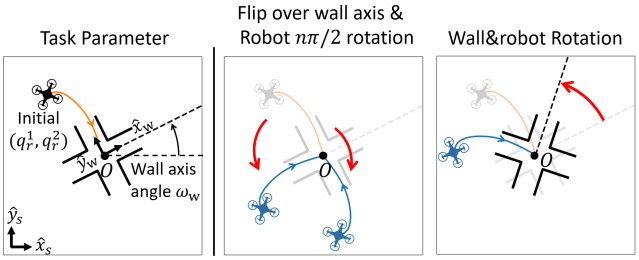

Figure 4: *Left*: The task parameter of 2D mobile robot's goal reaching task. *Right*: Symmetry transformations.

Construction of $\bar{h}$: Figure 5 visualizes the equivaraint module $\bar{h}(\tau) = (\bar{h}_1(\tau), \bar{h}_2(\tau))$. $\bar{h}_1(\tau) \in p4m$ is determined based on the robot's position relative to the wall state as shown in Figure 5 (*Left*). $\bar{h}_2(\tau) \in \mathrm{SO}(2)$ is determined based on the wall axis angle ω_w (see Figure 5 *Right*). This module $\bar{h}(\tau)$ is equivariant to $p4m \times \mathrm{SO}(2)$; more details including the proof are in Appendix C.2.1.

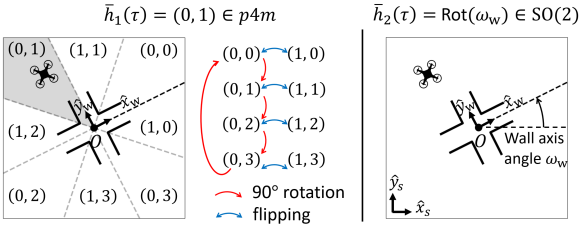

Figure 5: Illustration of the $(p4m, \mathrm{SO}(2))$-equivariant module $\bar{h}(\tau)$.

Assuming a one-dimensional latent space, we train fully connected autoencoders for MMPs and EMMPs, and TC-VAE where temporal convolutional neural networks are used as in [37]. Table 1 shows the four evaluation metrics, where the success rate is measured as follows: (i) we sample (τ, z) from the uniform distribution on $\mathcal{T}$ and learned density $p(z)$, (ii) generate trajectories via $f(z, \tau)$, and (iii) the generated trajectory is considered successful if it is consistent with the task parameter and reaches the goal without colliding to the wall. More details are in Appendix C.2.2.

Table 1: Reconstruction Error (RE), Mutual Information (MI), and Negative Log-Likelihood (NLL); the lower, the better. Success Rate (SR); the higher, the better.

Method	RE (↓)	MI (↓)	NLL (↓)	SR (↑)
TC-VAE [37]	0.257	0.268	1.50×10^4	53.18%
MMP	0.223	0.487	1.49×10^4	50.08%
MMP + indep	0.225	0.329	1.48×10^4	52.39%
EMMP	0.223	0.082	1.25×10^4	92.40%
EMMP + indep	0.229	0.077	1.24×10^4	86.66%

First of all, the EMMP methods show much higher success rates than both the MMP methods and TC-VAE, which is in large part attributed to the independence between z and τ and latent density fitting as shown in Table 1. As observed from the MI scores of the MMP + indep (0.329) and EMMP (0.082), using invariant and equivariant mappings is much more effective than the independence regularization term at making z and τ independent, which consequently makes learning $p(z)$ easier (see Figure 6).

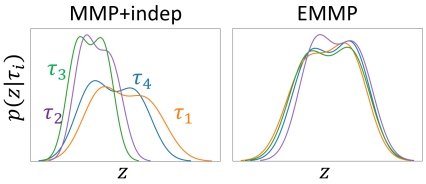

Figure 6: $p(z|\tau)$ of the MMP + indep and EMMP.

Second, the success rates of MMPs and TC-VAE do not show a noticeable difference. This suggests that the network architecture (temporal CNN or FCN) and autoencoder method (VAE or AE) do not have a significant impact on performance. A further comparison using EMMP in this regard is provided in Appendix C.2.3. Third, while the independence regularization term slightly improves the success rate of MMP, for EMMP, it rather shows a negative effect. Given the already low MI of EMMP, further reducing it at the expense of RE appears to have negatively impacted the success rate.

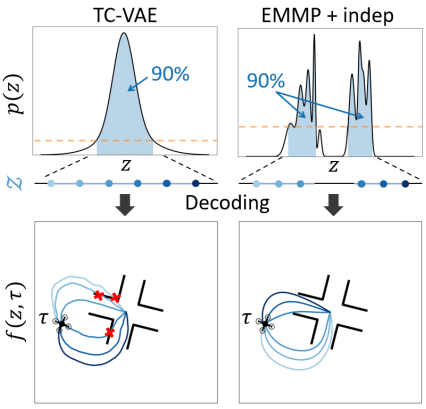

Figure 7: *Top*: $p(z)$. *Bottom*: Generated trajectories.

Figure 7 (*Top*) visualizes learned latent space densities $p(z)$ of TC-VAE and EMMP + indep. We select equally-spaced 6 latent points $\{z_i\}_{i=1}^6$ in $\mathcal{Z}$ for each model as shown in Figure 7 (blue points). The trajectories are then generated by the decoders as shown in Figure 7 (*Bottom*). The EMMP + indep shows a much higher success rate than the TC-VAE; the support of $p(z)$ in our method has two connected components, each corresponding to the set of trajectories that passes through one of the two closest passages to the robot.

4.2 Water-Pouring Task of a Franka Panda Robot

In this section, we consider a water-pouring task for a Franka Emika Panda robot arm; the robot is assumed to hold the bottle initially upright and is required to pour 150g of water into the cup (see Figure 8 *Left*). As shown in Figure 8 *Right*, a human demonstrator provides multiple water-pouring trajectories, which are assumed to form a continuous manifold. Our goal is to learn the motion manifold primitive for an arbitrary cup position, a bottle's initial pose, and the amount of water in the bottle. In particular, depending on the amount of water in the bottle, the demonstration trajectories have very different characteristics (see Figure 15 *Left* in Appendix C.3.3).

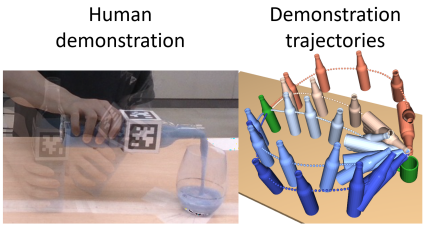

Figure 8: Demonstration data.

The configuration space is the space of the bottle pose, $\mathcal{Q} = \mathrm{SE}(3)$, the trajectory space is $\mathcal{X} = \mathrm{SE}(3)^T$. The task parameter τ consists of the cup position on the table $(q_c^1, q_c^2) \in \mathbb{R}^2$, the bottle's initial pose $T_b \in \mathrm{SE}(3)$, and the mass of the water $m_w \in [0.2, 0.41]$ (See Figure 9 *Left*). We collect 5 demonstration trajectories for 35 different task parameters in a total of 175 trajectories (details are included in Appendix C.3.2). The demonstration trajectories are collected by recording a video of a human demonstrator performing the water-pouring task and extracting the bottle's SE(3) trajectories using AprilTag [43]. Figure 8 *Right* shows 5 trajectories demonstrated for a given single task parameter τ. More details on the task parameter selection, data generation, and data split for training, validation, test are included in Appendix C.3.2.

Symmetries: There exist three symmetries that preserve the relative positional and geometric relationship between the cup and the bottle: (i) translation of the cup and the bottle by the same amount, (ii) rotation of the bottle around the cup, and (iii) rotation of the bottle around itself (see Figure 9 *Right*). The symmetry transformations can be described as group actions with the symmetry group $H := \mathbb{R}^2 \times \mathrm{SO}(2) \times \mathrm{SO}(2)$. Details on the group operations in H and group actions on $\mathcal{T}$ and $\mathcal{X} \times \mathcal{T}$ are discussed in Appendix C.3.2.

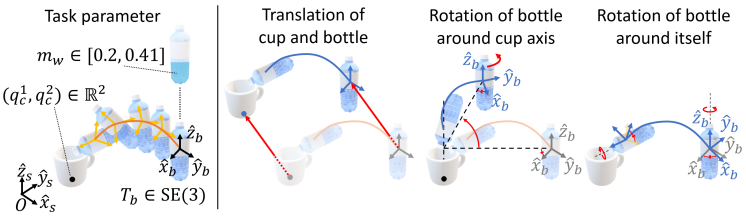

Figure 9: *Left*: The task parameter of water-pouring task. *Right*: Symmetry transformations.

Definition of H and $\bar{h}$: We define $\bar{h}(\tau) = (\bar{h}_1(\tau), \bar{h}_2(\tau), \bar{h}_3(\tau))$ as shown in Figure 10. $\bar{h}_1(\tau)$ is determined as the cup's position as shown in Figure 10 *Left*. $\bar{h}_2(\tau)$ is determined based on the position of the bottle relative to the cup (see Figure 10 *Middle*). $\bar{h}_3(\tau)$ is determined based on the bottle's orientation relative to the base frame (see Figure 10 *Right*).

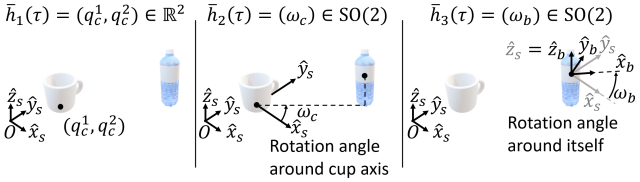

Figure 10: $(\mathbb{R}^2, \mathrm{SO}(2), \mathrm{SO}(2))$-equivariant module $\bar{h}(\tau)$.

Assuming a two-dimensional latent space dimension, we train EMMP + indep with fully connected neural networks and TC-VAE. Table 2 shows the four evaluation metrics (RE, MI, NLL, SR) – where the generated robot motion is considered successful if even a little water can be poured into the cup without spilling – and one additional metric that measures the error in the amount of poured water (150g water has been poured in demonstration data), which we refer to as the Water-Pouring Error (WPE). To generate motions given a task parameter τ from $\mathcal{T}$, we sample z from the learned density $p(z)$ and generate the bottle's trajectory via $f(z, \tau)$. If the generated SE(3) trajectory is out of the robot's workspace, i.e., the inverse kinematics solution does not exist, then we re-sample z

until we obtain a feasible trajectory. We measure the SR and WPE with 4 task parameters each with 5 samples, where we run a total of 20 generated trajectories on the real Panda robot.

As shown in Table 2, the EMMP + indep significantly outperforms TC-VAE. The large margin in RE and MI leads to lower NLL and a much higher task success rate of EMMP. Out of 20 trials of the TC-VAE, 6 trials fail to pour water into the cup and 5 trials spill water – where one trial both fails to pour water into the cup and spills –, resulting in only 10 successful pourings. On the other hand, EMMP + indep results in a 100% success rate. In addition, the WPE in EMMP + indep is also much lower than that of TC-VAE. While the EMMP's WPE (36.75g) out of 150g seems high, we note that this error is not caused by the manifold primitive learning error, but rather is attributed to the slippage between our two-finger gripper and the bottle. Even when we replay the demonstration trajectories on the robot, the water pouring error exists and it is 24.7g on average, implying 36.75g error is not that high.

Table 2: Reconstruction Error (RE), Mutual Information (MI), Negative Log-Likelihood (NLL), and Water-Pouring Error (WPE); the lower, the better. (SR); the higher, the better.

Method	TC-VAE	EMMP + indep
RE (↓)	0.183	0.129
MI (↓)	0.758	0.081
NLL (↓)	1.18×10^5	6.40×10^4
SR (↑)	$10/20$	$20/20$
WPE (↓)	87.2 ± 55.6	36.8 ± 18.5

To show the strong adaptability of our framework, we perform an obstacle avoidance task using EMMP + indep. Suppose there is an obstacle, not seen during training, that blocks some water-pouring trajectories in the learned manifold primitives (e.g., Figure 11 *Left*). Since we have learned a motion manifold, not a single trajectory, even if some trajectories are blocked, we can easily find an alternative collision-free trajectory from the learned manifold primitives as shown in Figure 11 *Right*. More details on the collision detection and obstacle avoidance algorithms are in Appendix C.3.2

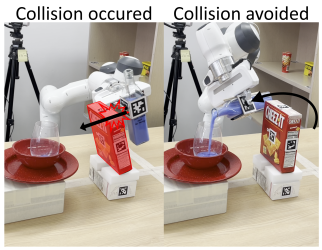

Figure 11: Obstacle avoidance.

5 Limitations

One of the key assumptions in our framework is the homeomorphic manifold assumption, that is $\mathcal{M}_\tau$ for all $\tau \in \mathcal{T}$ are homeomorphic, which may not hold depending on the problem. In such cases, instead of $p(z)$, we need to fit $p(z|\tau)$, which would require more demonstration data. Second, to construct an invariant encoder and equivariant decoder, we need to define an equivariant module $\bar{h}$ for a given symmetry group H. Although, in our case studies, it is relatively straightforward to construct $\bar{h}$, this process may not be trivial or even impossible depending on the problem. Lastly, as our experimental results show, the independence regularization term itself is not sufficient to enforce independence between z and τ. Finding a better independence regularization method would be an important future research direction.

6 Conclusion

In this paper, we have proposed a new family of highly adaptable movement primitive models, motion manifold primitives – which is a set of trajectory manifolds $\{\mathcal{M}_\tau\}$ for all task parameters τ –, and an autoencoder-based framework for learning them. To tackle the challenges in learning $\mathcal{M}_\tau$ such as requiring many demonstration data, (i) under the homeomorphic manifold assumption, we develop the motion manifold primitives framework and introduce the independence regularization term – where we enforce independence between z and τ so that it is sufficient to learn $p(z)$ instead of $p(z|\tau)$ – and (ii) we propose equivariant motion manifold primitives for an arbitrary symmetry group H in the robot task and a method to parameterize it by constructing an invariant encoder and equivariant decoder. Extensive experiments have confirmed the strong adaptability of our framework and that the equivariant manifold modeling is highly effective at learning accurate $\mathcal{M}_\tau$, which leads to superior performance compared to the existing method by a significant margin.

References

[1] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3: 297–330, 2020.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[3] Z. Zhu and H. Hu. Robot learning from demonstration in robotic assembly: A survey. *Robotics*, 7(2):17, 2018.

[4] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel. Dynamic movement primitives in robotics: A tutorial survey. *arXiv preprint arXiv:2102.03861*, 2021.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[6] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Trajectory formation for imitation with nonlinear dynamical systems. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 2, pages 752–757. IEEE, 2001.

[7] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning rhythmic movements by demonstration using nonlinear oscillators. In *Proceedings of the ieee/rsj int. conference on intelligent robots and systems (iros2002)*, number CONF, pages 958–963, 2002.

[8] S. Schaal, P. Mohajerian, and A. Ijspeert. Dynamics systems vs. optimal control—a unifying view. *Progress in brain research*, 165:425–445, 2007.

[9] A. Pervez, A. Ali, J.-H. Ryu, and D. Lee. Novel learning from demonstration approach for repetitive teleoperation tasks. In *2017 IEEE World Haptics Conference (WHC)*, pages 60–65. IEEE, 2017.

[10] Y. Fanger, J. Umlauft, and S. Hirche. Gaussian processes for dynamic movement primitives with application in knowledge-based cooperation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3913–3919. IEEE, 2016.

[11] J. Umlauft, Y. Fanger, and S. Hirche. Bayesian uncertainty modeling for programming by demonstration. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6428–6434. IEEE, 2017.

[12] A. Pervez, Y. Mao, and D. Lee. Learning deep movement primitives using convolutional neural networks. In *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*, pages 191–197. IEEE, 2017.

[13] S. Bahl, M. Mukadam, A. Gupta, and D. Pathak. Neural dynamic policies for end-to-end sensorimotor learning. *Advances in Neural Information Processing Systems*, 33:5058–5069, 2020.

[14] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[15] K. Neumann, A. Lemme, and J. J. Steil. Neural learning of stable dynamical systems based on data-driven lyapunov candidates. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1216–1222. IEEE, 2013.

[16] S. M. Khansari-Zadeh and A. Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6): 752–765, 2014.

[17] A. Lemme, K. Neumann, R. F. Reinhart, and J. J. Steil. Neural learning of vector fields for encoding stable dynamical systems. *Neurocomputing*, 141:3–14, 2014.

[18] K. Neumann and J. J. Steil. Learning robot motions with stable dynamical systems under diffeomorphic transformations. *Robotics and Autonomous Systems*, 70:1–15, 2015.

[19] C. Blocher, M. Saveriano, and D. Lee. Learning stable dynamical systems using contraction theory. In *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 124–129. IEEE, 2017.

[20] N. Figueroa and A. Billard. A physically-consistent bayesian non-parametric mixture model for dynamical system learning. In *CoRL*, pages 927–946, 2018.

[21] V. Sindhwani, S. Tu, and M. Khansari. Learning contracting vector fields for stable imitation learning. *arXiv preprint arXiv:1804.04878*, 2018.

[22] J. Z. Kolter and G. Manek. Learning stable deep dynamics models. *Advances in neural information processing systems*, 32, 2019.

[23] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters. Active incremental learning of robot movement primitives. In *Conference on Robot Learning*, pages 37–46. PMLR, 2017.

[24] N. Jaquier, D. Ginsbourger, and S. Calinon. Learning from demonstration with model-based gaussian process. In *Conference on Robot Learning*, pages 247–257. PMLR, 2020.

[25] M. Schneider and W. Ertel. Robot learning by demonstration with local gaussian process regression. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 255–260. IEEE, 2010.

[26] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9:1–29, 2016.

[27] D. A. Duque, F. A. Prieto, and J. G. Hoyos. Trajectory generation for robotic assembly operations using learning by demonstration. *Robotics and Computer-Integrated Manufacturing*, 57: 292–302, 2019.

[28] C. Yang, C. Chen, N. Wang, Z. Ju, J. Fu, and M. Wang. Biologically inspired motion modeling and neural control for robot learning from demonstrations. *IEEE Transactions on Cognitive and Developmental Systems*, 11(2):281–291, 2018.

[29] S. Chernova and M. Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, 2007.

[30] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[31] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell. Kernelized movement primitives. *The International Journal of Robotics Research*, 38(7):833–852, 2019.

[32] T. Osa, A. M. G. Esfahani, R. Stolkin, R. Lioutikov, J. Peters, and G. Neumann. Guiding trajectory optimization by demonstrated distributions. *IEEE Robotics and Automation Letters*, 2(2):819–826, 2017.

[33] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 91–98. IEEE, 2008.

[34] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In *2009 IEEE international conference on robotics and automation*, pages 2587–2592. IEEE, 2009.

[35] S. M. Khansari-Zadeh and A. Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32:433–454, 2012.

[36] M. Ginesi, D. Meli, A. Calanca, D. Dall'Alba, N. Sansonetto, and P. Fiorini. Dynamic movement primitives: Volumetric obstacle avoidance. In *2019 19th international conference on advanced robotics (ICAR)*, pages 234–239. IEEE, 2019.

[37] M. Noseworthy, R. Paul, S. Roy, D. Park, and N. Roy. Task-conditioned variational autoencoders for learning movement primitives. In *Conference on robot learning*, pages 933–944. PMLR, 2020.

[38] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[39] Y. Lee, H. Kwon, and F. Park. Neighborhood reconstructing autoencoders. *Advances in Neural Information Processing Systems*, 34:536–546, 2021.

[40] Y. Lee, S. Yoon, M. Son, and F. C. Park. Regularized autoencoders for isometric representation learning. In *International Conference on Learning Representations*, 2022.

[41] C. Jang, Y. Lee, Y.-K. Noh, and F. C. Park. Geometrically regularized autoencoders for non-euclidean data. In *The Eleventh International Conference on Learning Representations*.

[42] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[43] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE, 2011.