



Article

# Probabilistic Dual-Space Fusion for Real-Time Human-Robot Interaction

Yihui Li <sup>1,2,†</sup> , Jiajun Wu <sup>1,2,†</sup> , Xiaohan Chen <sup>1,2</sup> and Yisheng Guan <sup>1,2,\*</sup>

<sup>1</sup> Biomimetic and Intelligent Robotics Lab (BIRL), Guangdong University of Technology, Guangzhou 510006, China; liyihui.ing@gmail.com (Y.L.); 2112101012@mail2.gdut.edu.cn (J.W.); 2112001215@mail2.gdut.edu.cn (X.C.)

<sup>2</sup> State Key Laboratory of Precision Electronic Manufacturing, Guangzhou 510006, China

\* Correspondence: ysguan@gdut.edu.cn

† These authors contributed equally to this work.

**Abstract:** For robots in human environments, learning complex and demanding interaction skills from humans and responding quickly to human motions are highly desirable. A common challenge for interaction tasks is that the robot has to satisfy both the task space and the joint space constraints on its motion trajectories in real time. Few studies have addressed the issue of hyperspace constraints in human-robot interaction, whereas researchers have investigated it in robot imitation learning. In this work, we propose a method of dual-space feature fusion to enhance the accuracy of the inferred trajectories in both task space and joint space; then, we introduce a linear mapping operator (LMO) to map the inferred task space trajectory to a joint space trajectory. Finally, we combine the dual-space fusion, LMO, and phase estimation into a unified probabilistic framework. We evaluate our dual-space feature fusion capability and real-time performance in the task of a robot following a human-handheld object and a ball-hitting experiment. Our inference accuracy in both task space and joint space is superior to standard Interaction Primitives (IP) which only use single-space inference (by more than 33%); the inference accuracy of the second order LMO is comparable to the kinematic-based mapping method, and the computation time of our unified inference framework is reduced by 54.87% relative to the comparison method.

**Keywords:** dual-space fusion; physical human-robot interaction; real-time HRI; probabilistic learning; robot learning; learning from demonstrations; imitation learning



**Citation:** Li, Y.; Wu, J.; Chen, X.; Guan, Y. Probabilistic Dual-Space Fusion for Real-Time Human-Robot Interaction. *Biomimetics* **2023**, *8*, 497. <https://doi.org/10.3390/biomimetics8060497>

Academic Editor: Shuai Li

Received: 1 August 2023

Revised: 27 September 2023

Accepted: 10 October 2023

Published: 19 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As robotics advance, human-robot integration and human-robot interaction (HRI) have emerged as new research directions. To enable robots to seamlessly integrate into human daily life, it is essential to study HRI technology, especially physical human-robot interaction (pHRI) [1]. One of the main research challenges is how to make robots learn the skills to interact with humans more effectively. A popular research topic is to make robots learn collaborative skills through an imitative learning approach. This approach allows robots to generalize a skill or task by imitating it, thus avoiding complex expert programming [2,3]. Common imitation learning methods include the Hidden Markov Model (HMM) [4], the Gaussian Mixture Model (GMM) [5], the Dynamical Movement Primitives (DMP) [6], the Probabilistic Movement Primitives (ProMP) [7], and the Kernelized Movement Primitives (KMP) [8], etc.

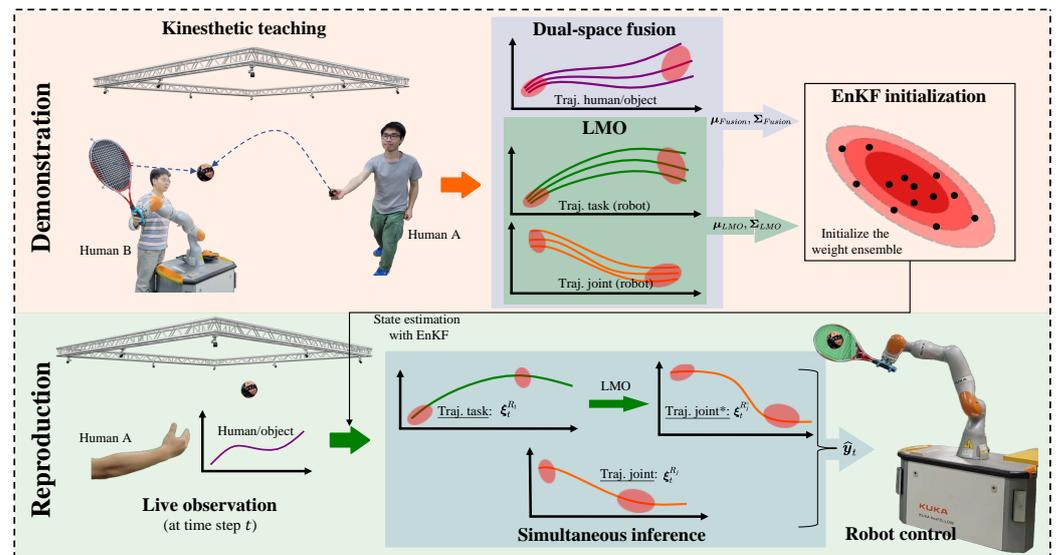
Learning skills from demonstration samples is similar to a martial arts practitioner imitating a master's movements, and a proficient student can acquire various features from the master's demonstrations. Likewise, researchers expect robots to learn more features from the demonstration samples. To address robot skill learning, researchers have proposed the concept of dual-space feature learning [9], where dual space usually refers to the task space and the joint space of the robot. Task space is a description of robot

actions in Cartesian coordinates, and joint space is a description of robot actions in terms of joint angles. To perform complex and demanding tasks, the robot needs to learn different characteristics of skillful actions from both task space and joint space perspectives [10–12]. For example, when a robot imitates a human to generate the motion of writing brush characters, it should not only match the end-effector trajectory to the writing specification, but also adjust the joint posture to resemble the human's. Another example is when the robot operates in a dynamic environment where obstacles or targets may change their positions or orientations. By integrating the learned knowledge of both task space and joint space from demonstrations, the robot can adapt its motion to the changing situation and avoid collisions or reach the desired goal. Therefore, the learning algorithm should extract and fuse both end-effector and joint posture features from the demonstrated action.

These requirements are not only in the field of skill imitation learning but also in the field of HRI where it is desirable for the robot to learn more action features. In our previous work [13], we proposed a method for simultaneous generalization and probabilistic fusion of dual-space trajectories, which enables the robot to simultaneously satisfy the dual-space constraints in an HRI task. We demonstrated the feasibility of the method through experiments in which UR5 followed a human-held object. This case required that the robot end-effector followed the target object in the task space, whereas the joint pose needed to move in a specific pattern so as to bypass the box next to it. However, the approach relies on the robot model and requires high computation of the robot Jacobian matrix for each time step. Therefore, in this paper, we extend our previous work and propose a method that does not rely on a robot model, making it more general. In addition, the learning of multiple features is essentially the establishment and inference of a multidimensional model, and with an increase in model dimensionality, the computational performance of the algorithm will be affected to some extent. Therefore, this paper carries out further research and enhancement on the real-time performance of the HRI method. The overview of the proposed framework is shown in Figure 1. Furthermore, our contributions are mainly threefold:

- A physical human-robot interaction method with dual-space (task space and joint space) features fusion is proposed, which improves the accuracy of the inferred trajectories in both spaces.
- We also propose a data-driven linear mapping operator for mapping from task space to joint space, which does not rely on robot models and time-consuming Jacobian computation, and is suitable for HRI scenarios especially those involving high-dimensional modeling.
- We present a unified probabilistic framework for integrating dual-space trajectories fusion, linear mapping operator and phase estimation, which scales particularly well with high dimensionality of the task and reduces the inference computation.

The rest of the paper is organized as follows. We begin by reviewing the previous related work on motion generation methods and spaces fusion in the field of HRI in Section 2. In Sections 3–5, we outline the methodology of our framework including dual-space fusion, linear mapping operator, phase estimation, and the unified framework. We review and discuss our experimental setups and results in Section 6. Finally, we conclude our work in Section 7.



**Figure 1.** Overview of the proposed method. **Top:** During the demonstrations, human A performs some actions (such as throwing a ball in the figure), human B teaches the robot to respond to Human A's actions through kinesthetic teaching, and the motion capture system synchronously records the task space and joint space trajectories of the robot, as well as the task space trajectories of Human A or object (for example, in the hitting experiment (Section 6.4), the task space trajectory of the ball thrown by Human A is recorded, and in the object following experiment (Section 6.1.2), the task space trajectory of the human-held object is recorded). At the **dual-space fusion** (Section 3) block, the demonstrations (all the trajectories) are decomposed with several Gaussian basis functions and obtain parameters of the weight distribution; at the **LMO** (Section 4) block, the mapping from task space to joint space is linearized by Taylor expansion, and then the parameters of its series are solved by Gaussian distribution, obtaining the parameters of the distribution. All the distribution parameters are transformed into a fusion ensemble of samples (right). **Bottom:** during live interaction, the robot generates a response trajectory (right) to the observation of human A/object with the learned fusion ensemble and LMO (Section 5).

## 2. Related Work

The primary problem in pHRI is generating robot motion by observing human motion. Amor et al. [14] proposed Interaction Primitives (IP) based on DMP for modeling the relationship between human and robot. IP is to fit the human trajectory and the robot trajectory separately using the DMP method, then combine the DMP parameters of the two trajectories, and finally observe the human trajectory and infer and predict the robot trajectory. IP has been applied to multiple scenarios [15,16]. However, this method only models the spatial uncertainty in the human-robot interaction process and does not model the temporal uncertainty. Therefore, it is difficult to avoid the problem of time synchronization in its interaction process. At the same time, methods based on the DMP framework need to select the type and number of basis functions. As the dimension increases, the number of basis functions also increases, and the complexity of the inference calculation process also increases. Huang et al. proposed KMP [8] to take the human motion sequence as the input state and directly predict the robot trajectory. Similar to the KMP method, Silvério et al. [17] performed HRI through Gaussian Process (GP), which also avoided the problem of time synchronization, but this method did not consider the covariance and model generalization of multi-dimensional trajectories. Vogt et al. [18,19] used GMM and HMM to model the interaction between humans and humans, and proposed the Interactive Meshes (IMs) method, which mapped the spatiotemporal relationship of teaching motion to the interaction between humans and robots. Wang et al. [20] utilized a Gaussian Process (GP) dynamics model to infer human intention based on human behavior and applied it to ping-pong robots. Ikemoto et al. [21] applied GMM to the scenario of physical contact

between humanoid robots and humans. Although these works use many mature theories to solve the problems of temporal and spatial uncertainty involved in robot learning, they almost all consider a single spatial feature, either joint space or task space, without involving the fusion of both. Therefore, when the task involves multiple spatial requirements, such as considering joint avoidance during robot trajectory tracking [22,23], these methods are not competent.

Dual-space trajectories fusion refers to learning the movement features of different spaces in demonstrations by fusing the movements of a robot's task space and joint space, and the purpose is to complete the robot operation tasks under the joint constraints of both spaces. In the field of HRI, most current works only focus on the single space of robotics, e.g., Chen et al. proposed an HRI framework in joint space without constraining the task space [24]. Li et al. [25] proposed a topology-based motion generalization method for the general motion generation problem, which maintains the spatial relationship between different parts of a robot or between different agents with a topology representation; the configuration prior (joint space) is preserved in the generated motion, and the target position (task space) of the end-effector also can also be realized by applying certain constraint. However, the process of generating these two space features is separative and there is no fusion essentially. Calinon et al. [22] used the relative distance trajectory between the robot and the object as the task space trajectory and modeled it. Specifically, the task space trajectory and joint space trajectory of the robot in the demonstrations are modeled with GMM separately. Then Gaussian Mixture Regression (GMR) is leveraged to obtain the probability distribution of the two spatial trajectories, and finally Gaussian features are used to fuse the two trajectories. Schneider et al. [26] adopted a similar way to perform dual-space feature learning of robot trajectories, but replaced the modeling method of GMM and GMR with Heteroscedastic Gaussian Processes (HGP) [27,28]. However, Calinon and Schneider's work did not perform joint modeling of the two spatial trajectories, and could not achieve synchronous generalization of the two spaces. Its essence is still to generalize the joint space trajectory. In addition, their method cannot guarantee that the generalization directions of different spatial trajectories are consistent, which means that the generalization result for one space may not be beneficial to another space. To this end, Huang et al. [29] presented a work for solving the problem of dual-space synchronous generalization. They considered the generalization of task space while using null space to generalize robot joint space trajectory, but they did not consider the problem of interaction between humans and robots, and could not be used for real-time interaction scenarios.

Regarding the real-time performance of those methods, KMP can model the trajectory uncertainty outside the range of the demonstrations, and the kernel function processing method in the KMP inference process is similar to the Heteroscedastic Gaussian Process, its computational complexity is  $\mathcal{O}(N^3)$ . It cannot adapt well in the case of long trajectory sequences and has poor real-time performance. Campbell et al. proposed Bayesian Interaction Primitives (BIP) [30] and the improved BIP [31] inspired by the related method of robot pose estimation and update in Simultaneous Localization and Mapping (SLAM). Specifically, BIP is essentially based on the work of Amor et al. [14] and Maeda et al. [32], and applies Extended Kalman Filter (EKF) to the phase estimation of Interaction Primitives, making the phase estimation of Interaction Primitives and Bayesian inference [33] process synchronized, and improving computational efficiency. The time complexity of the time alignment method in the original IP framework is  $\mathcal{O}(N^2)$ , BIP removes the extra time alignment process, reducing the interaction inference time by 66%.

As shown in Table 1, most of the existing research on pHRI or skill learning focuses on either task space or joint space, but not both. Moreover, few methods can achieve dual-space fusion and generalization, as well as meet the real-time interaction requirements. However, the existing methods have advanced in modeling spatial and temporal uncertainties, phase estimation, and other related theories, which can provide guidance for our further exploration.

**Table 1.** Related works comparison in required terms.

Method or Author	HRI	Spatial Uncertainty Modeling	Time Uncertainty Modeling	Dual-Space Fusion	Meet Real-Time Interaction Requirements
GMM/GMR [5]	✗	✓	✗	✗	✗
HMM/HSMM [4]	✗	✓	✗	✗	✗
DMP [6]	✗	✗	✗	✗	✗
ProMP [7]	✗	✓	✗	✗	✗
GP [34]	✗	✗	-	✗	✗
KMP [8]	✓	✓	-	✗	✗
IP [14]	✓	✓	✗	✗	✓
BIP [30]	✓	✓	✓	✗	✓
TMG [25]	✗	✓	✗	✗	✗
Calinon et al. [22]	✗	✓	✗	✗	✗
Schneider et al. [26]	✗	✓	✗	✗	✗
Huang et al. [29]	✗	✓	✗	✓	✗
Requirement	✓	✓	✓	✓	✓

✓: yes, ✗: no, -: unnecessary.

### 3. Preliminaries: Probabilistic Fusion in Task Space and Joint Space

#### 3.1. Simultaneous Generalization of Dual Space

The goal of the simultaneous generalization of dual space is to enable the robots (the controlled agent in pHRI) to learn the features of the motion trajectories in dual space. This enables the robot to perform complex tasks, such as tracking a target object while avoiding obstacles. To achieve the simultaneous generalization of dual space, two components are needed: the learning phase and the inference phase. The former aims to learn the features of trajectories in dual space. The latter aims to generate the robot’s motion trajectories in dual space based on the observed human motion trajectories.

In the learning phase, we propose a probabilistic fusion model of task space and joint space, inspired by IP [14]. This model encodes the human and robot motion trajectories with basis functions and models their spatial and temporal uncertainty with probabilistic weights. We define the joint trajectory of human and robot  $f_{1:T} = [\zeta_{1:T}^H, \zeta_{1:T}^R]^T$  as a sequence of sensor observations of length  $T$ . Here,  $\zeta_{1:T}^H$  is a sequence of human movement trajectories with  $D_H$  degrees of freedom. For a time step  $t$ , we denote  $\zeta_t^H = [y_1^H|_t, y_2^H|_t, \dots, y_{D_H}^H|_t]$  as the human trajectory at that time step, or simply  $\zeta_t^H = [y_1^H, y_2^H, \dots, y_{D_H}^H]$  for brevity.  $\zeta_t^R = [\zeta_t^{R_t}, \zeta_t^{R_j}]$  is a sequence of robot movement trajectories with the number of degrees of freedom  $D_R = D_{R_t} + D_{R_j}$  at time step  $t$ .  $\zeta_t^{R_t}$  is the trajectory point of the robot in task space at time step  $t$ ,  $\zeta_t^{R_j}$  is the trajectory point of the robot in joint space, and  $D_{R_t}$  and  $D_{R_j}$  are the numbers of degrees of freedom of the two space trajectories.

The sequence for each degree of freedom is linearly fitted using  $B$  basis functions containing Gaussian white noise. This fitting method is derived from DMP:

$$y_d = \Phi^T(t)w_d + \varepsilon_\zeta \tag{1}$$

where  $\Phi^T(t)$  is the vector composed of  $B$  Gaussian basis functions,  $w_d$  is the weight corresponding to the basis function, and  $\varepsilon_\zeta$  is the corresponding Gaussian white noise.

The joint weight matrix  $w = [w^H, w^{R_t}, w^{R_j}]^T$  is obtained by fitting using Equation (1). It is assumed that the weights learned from each demo follow a Gaussian distribution, i.e.,  $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ , and the Gaussian parameters  $\mu_w \in \mathbb{R}^{DB \times 1}$  and  $\Sigma_w \in \mathbb{R}^{DB \times DB}$  can be obtained from multiple demos by Maximum Likelihood Estimation (MLE), where  $D$  is the sum of the dimensions of the human movement trajectories and the robot dual space movement trajectories, and  $B$  is the number of Gaussian basis functions. Furthermore, it is important to note that  $\Sigma_w$  captures the correlation among human motion trajectories, robot motion trajectories in task space, and robot motion trajectories in joint space.

In the inference process of HRI, we use the weights learned from the learning phase as prior information. We observe the human motion trajectory  $\zeta_{1:t^*}^H$  from the begin-

ning of the interaction to time  $t^*$  and update the entire weight matrix using Bayesian recursive inference.

$$p(w|\zeta^*) \propto p(\zeta^*|w)p(w) \tag{2}$$

Then, we use the  $w^{R_t}$  and  $w^{R_j}$  components of the inferred weight matrix  $w$  to generate the robot motion trajectories in dual space according to Equation (1). So for an observation time step  $t^*$ , we denote the generated task space trajectory as  $\zeta_{t^*:T}^{R_t}$ , and the joint space trajectory as  $\zeta_{t^*:T}^{R_j}$ , where  $R_t$  and  $R_j$  represent robot task space and robot joint space, respectively.

### 3.2. Kinematics-Based Linear Mapping

We have obtained two control trajectories for the robot in both task space and joint space until now, but, in fact, we only need one controller to control the motion of the robot. Therefore, a mapping from task space to joint space for the trajectory is required, and then fusing the two joint space trajectories into one in the form of probability distributions. The most common mapping method in robotics is inverse kinematics with a kind of Jacobian matrix. So, based on the kinematic relationship of the robot, the task space trajectory sequence  $\zeta_{t^*:T}^{R_t}$  is linearly and iteratively mapped into the joint space.

Assume that the trajectory point  $\zeta_t$  at each time step  $t$  ( $t \in [t^*, T]$ ), follows a Gaussian distribution, i.e.,  $\zeta_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ , and obviously the parameters  $\mu_t$  and  $\Sigma_t$  can be obtained by performing a linear transformation (Equation (1)) on the updated weight distribution  $\mathcal{N}(\mu_w^*, \Sigma_w^*)$ , as well as the original and mapped joint space trajectory point  $\zeta_t^{R_j} \sim \mathcal{N}(\mu_t^{R_j}, \Sigma_t^{R_j})$  and  $\zeta_t^{R_j^*} \sim \mathcal{N}(\mu_t^{R_j^*}, \Sigma_t^{R_j^*})$ . The new trajectory obtained from the kinematics-based mapping is denoted as  $\zeta_{t^*:T}^{R_j^*}$ , and the update formula of a point of the new trajectory at time step  $t$  is:

$$\zeta_{t+1}^{R_j^*} = \zeta_t^{R_j^*} + J^\dagger \left( \zeta_t^{R_j^*} \right) \dot{\zeta}_t^{R_t} \tag{3}$$

$$\Sigma_{t+1}^{R_j^*} = \Sigma_t^{R_j^*} + J^\dagger \left( \zeta_t^{R_j^*} \right) \dot{\Sigma}_t^{R_t} J^\dagger \left( \zeta_t^{R_j^*} \right)^\top \tag{4}$$

where  $J^\dagger \left( \zeta_t^{R_j^*} \right)$  is the pseudo-inverse of the robot Jacobian matrix  $J \left( \zeta_t^{R_j^*} \right)$  at time step  $t$ , and  $\dot{\zeta}_t^{R_t}$  is the derivative of  $\zeta_t^{R_t}$ . Since  $\zeta_t^{R_j}$  follows a Gaussian distribution,  $\dot{\zeta}_t^{R_t}$  does so as well, i.e.,  $\dot{\zeta}_t^{R_t} \sim \mathcal{N}(\dot{\mu}_t^{R_t}, \dot{\Sigma}_t^{R_t})$ . In addition, the observed joint angle  $\zeta_{t^*}^{R_j}$  at the time step of  $t^*$  is taken as the initial value of the iteration, that is  $\zeta_{t^*}^{R_j^*} = \zeta_{t^*}^{R_j}$ .

### 3.3. Probabilistic Fusion

From the above iterative mapping method, two kinds of movement trajectories in robot joint space can be obtained, one of which is the robot joint space trajectory sequence  $\zeta_{t^*:T}^{R_j}$  obtained by direct extrapolation from the human action sequence  $\zeta_{1:t^*}^H$ , and the other is the joint space sequence  $\zeta_{t^*:T}^{R_j^*}$  mapped from the robot task space trajectory  $\zeta_{t^*:T}^{R_t}$ . At time step  $t$ , there are  $\zeta_t^{R_j} \sim \mathcal{N}(\mu_t^{R_j}, \Sigma_t^{R_j})$  and  $\zeta_t^{R_j^*} \sim \mathcal{N}(\mu_t^{R_j^*}, \Sigma_t^{R_j^*})$ .

$\zeta_{t^*:T}^{R_j}$  and  $\zeta_{t^*:T}^{R_j^*}$  can be regarded as two probability estimates obtained from the same space (joint space) for the same observation object (human motion sequence  $\zeta_{1:t^*}^H$ ). Therefore, the trajectory sequence  $\zeta_{t^*:T}^{R_j}$  and  $\zeta_{t^*:T}^{R_j^*}$  are two estimation results of the same estimation object (joint space control trajectory, denoted by  $\hat{\zeta}_j^R$ ), whereas  $\zeta_{t^*:T}^{R_j^*}$  contains the information of task space feature generalization,  $\zeta_{t^*:T}^{R_j}$  contains the information of joint space feature

generalization, and by fusing the two feature information at each time step, we can obtain the control trajectory that contains the features of both spaces.

The two estimates are considered to be independent of each other, and the distributions of the two trajectory estimates are fused at each time step  $t$  with Bayesian estimation:

$$p\left(\hat{\xi}_t^{R_j} \mid \xi_t^{R_j}, \xi_t^{R_j^*}\right) = \eta_t p\left(\hat{\xi}_t^{R_j} \mid \xi_t^{R_j}\right) p\left(\hat{\xi}_t^{R_j} \mid \xi_t^{R_j^*}\right) \tag{5}$$

$$\eta_t = \frac{p\left(\xi_t^{R_j}\right) p\left(\xi_t^{R_j^*}\right)}{p\left(\xi_t^{R_j}, \xi_t^{R_j^*}\right) p\left(\hat{\xi}_t^{R_j}\right)} \tag{6}$$

At time step  $t$ , the control trajectory of the robot in the joint space follows a Gaussian distribution  $\mathcal{N}\left(\hat{\mu}_t^{R_j}, \hat{\Sigma}_t^{R_j}\right)$ , with:

$$\hat{\mu}_t^{R_j} = \frac{\mu_t^{R_j} \Sigma_t^{R_j} + \mu_t^{R_j^*} \Sigma_t^{R_j^*}}{\Sigma_t^{R_j} + \Sigma_t^{R_j^*}} \tag{7}$$

Finally, we obtain the robot control trajectory  $\hat{\xi}_t^{R_j} = \hat{\mu}_t^{R_j}$ , which includes the fusion features of both task space and joint space.

#### 4. Kinematics-Free Linear Mapping Method

The mapping relationship between task space and joint space in the last subsection is derived from the robot kinematic model and relies on the Jacobian matrix computed from the robot model. We would like to make the application of the dual-space probabilistic fusion method in robot interaction skill learning more flexible and extend it to more non-kinematic scenarios. For example, we can map and fuse only a part of the interaction data in task space or joint space, or we can apply the method to other spatial fusion tasks that do not involve task space or joint space. However, the computation of the Jacobian matrix in inverse kinematics is time-consuming, so it is not suitable for HRI tasks with high real-time requirements. In this part, we will introduce a linear mapping operator (LMO) for this purpose.

##### 4.1. Linear Mapping Operators

We propose a linear mapping approach between two spaces based on Jacobian-like matrices, which are defined as operators that act on data (displacement) vectors. We generalize the notion of dual-space mapping beyond the conventional task and joint spaces and introduce two abstract function spaces, denoted as space  $A$  and space  $B$ . Assume that, given demonstration samples, we can learn a time-varying LMO denoted as  $\mathcal{L}_t$  for each time step  $t$  of HRI, such that it maps vectors from space  $A$  to space  $B$ , i.e.,  $A \rightarrow B$ .

A possible way to model and estimate the LMO ( $\mathcal{L}_t$ ) is to treat it as a dynamical system that varies over time. However, this approach may not capture the exact nature of the operator, because it may involve different mapping spaces and nonlinear or discrete transformations in the complex case of dynamical parameters. Therefore, we do not assume a specific mathematical model for the operator but rather regard it as a time-varying parameter to be estimated. There have been many studies on the estimation of time-varying system parameters, e.g., Cui et al. [35] used the Kalman Filter (KF) method to estimate the system state, Kim et al. [36] used the Extension Kalman Filter (EKF) method to estimate the nonlinear system, and Campbell et al. [30] applied the EKF to the trajectory weight inference of Interaction Primitives in HRI. We propose a framework using the Kalman estimation to jointly express the linear mapping operator and the robot interaction trajectory inference.

#### 4.2. Modeling of Linear Mapping Operators

Let  $\mathcal{L}$ ,  $\zeta^A$ , and  $\zeta^B$  be the time-dependent state sequences corresponding to  $\mathcal{L}_t$ ,  $\zeta_t^A$ , and  $\zeta_t^B$ , respectively. For an HRI process with a time length of  $T$ , the trajectories corresponding to  $\mathcal{L}_t$ ,  $\zeta_t^A$ , and  $\zeta_t^B$  can be denoted as  $\mathcal{L}_{0:T}$ ,  $\zeta_{0:T}^A$ ,  $\zeta_{0:T}^B$ , respectively. The mathematical relationship between  $\mathcal{L}_t$ ,  $\zeta_t^A$ , and  $\zeta_t^B$  is expressed as follows:

$$\zeta_t^A = \mathcal{L}_t \zeta_t^B, \quad t \in [0, T] \tag{8}$$

which denotes the trajectory point of space B mapped into space A at time step  $t$ , i.e.,  $\zeta_t^B \rightarrow \zeta_t^A$ , where  $\zeta_t^A \in \mathbb{R}^{m \times 1}$ ,  $\zeta_t^B \in \mathbb{R}^{n \times 1}$ ,  $\mathcal{L}_t \in \mathbb{R}^{m \times n}$ , where  $m$  and  $n$  is the number of dimensions of space A and space B, respectively, (a more specified definition for an HRI system is introduced in Section 3.1). The above equation can obtain:

$$\mathcal{L}_t = \left( \zeta_t^A \zeta_t^{A\top} \right) \left( \zeta_t^B \zeta_t^{B\top} \right)^\dagger \tag{9}$$

Although  $\mathcal{L}_t$  is the linear mapping relationship between two spaces, it is assumed to be a nonlinear dynamic system itself, and needs to be linearly approximated by Taylor expansion or Taylor series, and its expansion order affects the accuracy of model establishment. Theoretically, the higher the expansion order, the higher the mapping accuracy of the model, but it will also bring about a decrease in inference efficiency. Let  $o$  be the order of linear mapping operator expansion, then  $\mathcal{L}_t$  can be discretized and expressed as:

$$\mathcal{L}_t \approx \mathcal{L}_{t-1} + \frac{\partial \mathcal{L}_{t-1}}{\partial t} \Delta t + \frac{\partial \left( \frac{\partial \mathcal{L}_{t-1}}{\partial t} \right)}{\partial t} \frac{\Delta t^2}{2} + \dots + \frac{\partial \mathcal{L}_{t-1}^{o-1}}{\partial t} \frac{\Delta t^o}{o!}, \tag{10}$$

where  $\frac{\partial \mathcal{L}_{t-1}}{\partial t}$  and  $\frac{\partial \left( \frac{\partial \mathcal{L}_{t-1}}{\partial t} \right)}{\partial t}$  is the first and second partial derivative of  $\mathcal{L}_{t-1}$  with respect to  $t$ , respectively, marked as  $\mathcal{L}_t'$  (or  $\mathcal{L}_t^1$ ) and  $\mathcal{L}_t''$  (or  $\mathcal{L}_t^2$ ). At time step  $t$ , we assume that the coefficients of each term in the series follows a Gaussian distribution, i.e.,  $\mathcal{L}_{ij(t)} \sim \mathcal{N} \left( \mu_{\mathcal{L}_{ij(t)}}, \sigma_{\mathcal{L}_{ij(t)}}^2 \right)$ . Therefore, we can estimate the linear mapping operator by estimating the parameters of its finite expansion orders.

In practical applications, we need to learn from the demonstrations to obtain the parameters of the LMO. By observing the demonstration sample data at a certain observation frequency (120Hz in our demonstrations), we can obtain the human and robot motion data sequences in HRI. The LMO at each interaction time step in the demonstration sample is obtained by Equation (8). The initial value of the model is the mean of the initial LMO in the training sample.

At an observation frequency, the LMO of each sample is discretized at each time step  $t$ . For example, for the first two orders, we have  $\mathcal{L}_t' = \frac{\mathcal{L}_t - \mathcal{L}_{t-1}}{\Delta t}$ ,  $\mathcal{L}_t'' = \frac{\mathcal{L}_t' - \mathcal{L}_{t-1}'}{\Delta t}$ , where both  $\mathcal{L}_t'$  and  $\mathcal{L}_t''$  follows a Gaussian distribution, i.e.,  $\mathcal{L}_{ij(t)}' \sim \mathcal{N} \left( \mu_{\mathcal{L}_{ij(t)}'}, \sigma_{\mathcal{L}_{ij(t)}'}^2 \right)$ ,  $\mathcal{L}_{ij(t)}'' \sim \mathcal{N} \left( \mu_{\mathcal{L}_{ij(t)}''}, \sigma_{\mathcal{L}_{ij(t)}''}^2 \right)$ .

#### 4.3. Inference of Linear Mapping Operators

We adopt a nonlinear KF framework that follows the Bayesian estimation process to infer the LMO and the dual-space weights jointly. If only considering the fusion model and LMO, the state matrix of the model at time  $t$  is given by:

$$\mathbf{s}_t = [\ell_t^0, \ell_t^1, \dots, \ell_t^o, \mathbf{w}^\top]^\top, \quad \mathbf{s}_t \in \mathbb{R}^{D_w B + omn}, \tag{11}$$

where  $\ell_t^i$  is the vectorized form of  $\mathcal{L}_t^i$  ( $i$  is the order of the expansion,  $i \in [0, o]$ ), i.e.,  $\ell_t^i = [\mathcal{L}_{11(t)}^i, \dots, \mathcal{L}_{mn(t)}^i]^\top$ ;  $w$  is the joint weight of human and robot motions in two spaces, and  $w = [w^H, w^{R_A}, w^{R_B}]^\top$ ;  $D_w$  is the sum of the dimensions of the interaction primitive weights; and  $B$  is the number of basis functions.

The estimation process aims to compute the probability of the state matrix given an observation sequence  $z_{0:t}$ , and its conditional probability density function follows a Gaussian distribution  $\mathcal{N}(s_t | \mu_t, \Sigma_t)$ ; we have:

$$p(s_t | z_{0:t}) = \mathcal{N}(s_t | \mu_t, \Sigma_t), \tag{12}$$

$$\mu_0 = [\mu_{\ell_0}, \dots, \mu_{\ell_0^o}, \mu_w^\top]^\top, \tag{13}$$

$$\Sigma_0 = \begin{bmatrix} \Sigma_{\ell, \ell} & \Sigma_{\ell, w} \\ \Sigma_{w, \ell} & \Sigma_{w, w} \end{bmatrix}, \tag{14}$$

where  $\mu_t \in \mathbb{R}^{D_w B + omn}$  and  $\Sigma_t \in \mathbb{R}^{(D_w B + omn) \times (D_w B + omn)}$ . The observation object sequence is the HRI trajectory; that is,  $z_{0:t} = y_{0:t^*}$ ,  $\Sigma_{w, w} = \Sigma_w$ .

Bayesian filtering methodology consists of two components: the state transition model and the measurement model. Based on the linear approximation assumption of the linear mapping operator in Equation (10), we can establish the following state transition model for its vectorized form at each time step  $t$ :

$$\mu_{\ell_t} = G \mu_{\ell_{t-1}}, \tag{15}$$

$$\Sigma_{\ell_t} = G \Sigma_{\ell_{t-1}} G^\top + Q_{\ell_t}, \tag{16}$$

where

$$G = \begin{bmatrix} 1 & \Delta t & \dots & \frac{\Delta t^o}{o!} & 0 & \dots & 0 \\ 0 & 1 & \Delta t & \dots & \frac{\Delta t^o}{o!} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \frac{\Delta t^o}{o!} & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix},$$

$$Q_t = \begin{bmatrix} \sigma_{\mathcal{L}, \mathcal{L}}^2 & \sigma_{\mathcal{L}, \mathcal{L}'}^2 & \dots & \sigma_{\mathcal{L}, \mathcal{L}^o}^2 \\ \sigma_{\mathcal{L}, \mathcal{L}}^2 & \sigma_{\mathcal{L}, \mathcal{L}'}^2 & \dots & \sigma_{\mathcal{L}, \mathcal{L}^o}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{\mathcal{L}^o, \mathcal{L}}^2 & \sigma_{\mathcal{L}^o, \mathcal{L}'}^2 & \dots & \sigma_{\mathcal{L}^o, \mathcal{L}^o}^2 \end{bmatrix}.$$

And then we have the measurement model:

$$z_t = \underbrace{\begin{bmatrix} \Phi_\delta^\top w_1 \\ \Phi_\delta^\top w_2 \\ \vdots \\ \Phi_\delta^\top w_{D_w} \end{bmatrix}}_{h(\mu_t)} + \mathcal{N} \left( 0, \underbrace{\begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \sigma_{D_w}^2 \end{bmatrix}}_{R_t} \right), \tag{17}$$

and the partial derivative of  $h(\mu_t)$  with respect to the weight vector  $x_t = [w_1, w_2 \dots, w_{D_w}]^\top$  is:

$$H_t = \frac{\partial h(\mu_t)}{\partial x_t} = \begin{bmatrix} \frac{\partial \Phi_\delta^\top w_1}{\partial w_1} & \dots & \frac{\partial \Phi_\delta^\top w_1}{\partial w_{D_w}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial w_1} & \dots & \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial w_{D_w}} \end{bmatrix} = \begin{bmatrix} \Phi_\delta^\top & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \Phi_\delta^\top \end{bmatrix}. \tag{18}$$

Finally, by combining the measurement model and the state transition model with the Bayesian filtering methodology, the expansion parameters of the LMO are updated continuously by observing the trajectories of human/object.

### 5. Dual Space and Time Simultaneous Inference with EnKF

The previous two sections mainly focused on generating robot motions based on human motion trajectories; that is, modeling and inferring the spatial uncertainty of robot motion. Spatial uncertainty estimation is the process of inferring the motion state of the robot at each moment. By using the spatial uncertainty inference in the previous sections, we can obtain the robot’s motion trajectory at any interaction time step. However, the robot does not know which point on the trajectory it should move to at this time. Therefore, we need to model and infer the temporal uncertainty of robot motion, which is essentially modeling and estimating the phase factor of the robot’s current stage; that is, phase estimation [32].

We draw inspiration from the BIP method, which extends the Kalman Filter to perform spatiotemporal synchronization inference, and we propose an inference method that can infer the phase, the linear mapping operator, and the dual-space trajectory weights simultaneously.

#### 5.1. Spacetime Joint Modeling

In line with the modeling idea of this paper, the phase estimation problem can be formulated as how to find or estimate the value of  $\zeta(t) = \Phi(t)^\top w$  at the current time step  $t$ . We assume that the phase of the demonstration is a constant-speed change process, and for a sample of length  $T$ , its phase velocity  $\delta = 1/T$ , and let  $\beta$  be the average phase velocity obtained from the demonstrations.

We define the complete state matrix as follows, by concatenating the phase and phase velocity:

$$s_t = [\delta_t, \dot{\delta}_t, \ell_t, w^\top]^\top, s_t \in \mathbb{R}^{D_w B + omn + 2}, \tag{19}$$

where  $\delta_t$  and  $\dot{\delta}_t$  are the phase and phase velocity at time  $t$ , respectively;  $\ell_t$  is the parameter of the LMO;  $w^\top$  is the weight of the dual-space trajectories point. We initialize the phase estimation as follows:

$$\mu_0 = [0, \beta, \mu_\ell, \mu_w^\top]^\top. \tag{20}$$

To integrate phase estimation into the probabilistic inference of spatial trajectory, Equation (18) can be rewritten as:

$$\begin{aligned}
 H_t &= \begin{bmatrix} \frac{\partial \Phi_\delta^\top w_1}{\partial \delta} & \frac{\partial \Phi_\delta^\top w_1}{\partial \delta} & \frac{\partial \Phi_\delta^\top w_1}{\partial w_1} & \cdots & \frac{\partial \Phi_\delta^\top w_1}{\partial w_{D_w}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial \delta} & \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial \delta} & \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial w_1} & \cdots & \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial w_{D_w}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{\partial \Phi_\delta^\top w_1}{\partial \delta} & 0 & \Phi_\delta^\top & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \Phi_\delta^\top w_{D_w}}{\partial \delta} & 0 & 0 & \cdots & \Phi_\delta^\top \end{bmatrix}.
 \end{aligned} \tag{21}$$

The above process couples the phase estimation and the recursive state estimation of the dual-space trajectory weights. However, most of the existing studies use DTW to achieve phase estimation, which has a time complexity of  $\mathcal{O}(N^2)$  [14]; DTW needs to query and compare the entire observed trajectory sequence at each time step, so it requires extra computational overhead for phase estimation while performing trajectory inference. We combine phase estimation with state recursion, which can avoid this extra overhead; for the overall algorithm, the two phase factors ( $\delta_t, \dot{\delta}_t$ ) introduced actually only increase two

dimensions on the state matrix (Equation (19)), which has negligible impact on the overall state estimation dimension.

5.2. Spacetime Synchronization Inference

The linear mapping operator and the dual-space weights increase the dimension of the state matrix significantly. If we use the EKF inference method, we need to maintain a large covariance matrix at all times, and this high-dimensional matrix operation will bring more computational overhead. Therefore, we use the ensemble Kalman Filter (EnKF) inference method to handle the high-dimensional problem. EnKF does not need to maintain the covariance matrix like EKF, but uses an ensemble to simulate the state uncertainty.

The core idea of EnKF is as follows: we use an ensemble of dimension  $E$  to approximate the state prediction with the Monte Carlo method, where the sample mean of the ensemble is  $\mu$ , and the sample covariance matrix is  $\Sigma$ . Each element of the ensemble is obtained by the state transition model, and is perturbed by stochastic process noise:

$$x_{t|t-1}^j = Gx_{t-1|t-1}^j + \mathcal{N}(0, Q_t), \quad 1 \leq j \leq E, \tag{22}$$

where  $x_{t|t-1}$  denotes the prior of the state at time  $t$  obtained from time  $t - 1$ . When  $E$  approaches infinity, the ensemble will effectively simulate the covariance calculation result of Equation (16). In the scenarios of pHRI, each element in the ensemble comes from the demonstration in the training dataset. The state ensemble is transformed to the measurement space through the nonlinear observation equation, and the error between each element and the sample mean in the ensemble is calculated:

$$H_t X_{t|t-1} = [h(x_{t|t-1}^1), \dots, h(x_{t|t-1}^E)], \tag{23}$$

$$H_t A_t = H_t X_{t|t-1} - \left[ \frac{1}{E} \sum_{j=1}^E h(x_{t|t-1}^j), \dots, \frac{1}{E} \sum_{j=1}^E h(x_{t|t-1}^j) \right]. \tag{24}$$

Then, the new covariance matrix is:

$$S_t = \frac{1}{E-1} (H_t A_t) (H_t A_t)^\top + R_t. \tag{25}$$

The Kalman gain is calculated as follows:

$$A_t = X_{t|t-1} - \frac{1}{E} \sum_{j=1}^E x_{t|t-1}^j, \tag{26}$$

$$K_t = \frac{1}{E-1} A_t (H_t A_t)^\top S_t^{-1}. \tag{27}$$

The ensemble is updated and the measurement results with random noise are considered:

$$Y_t = [y_t + \epsilon_y^1, \dots, y_t + \epsilon_y^E], \tag{28}$$

$$X_{t|t} = X_{t|t-1} + K_t (Y_t - H_t X_{t|t-1}). \tag{29}$$

when  $\epsilon_y \sim \mathcal{N}(0, R_t)$ , the ensemble can accurately represent the error covariance of the optimal state estimation [37]. The measurement noise is calculated by the following:

$$R_t = \frac{1}{N} \sum_i \frac{1}{T_i} \sum_t^{T_i} (y_t - h([\delta_t, w_i]))^2. \tag{30}$$

Equations (22)–(29) describe the complete filtering process. In the scenarios of pHRI, we combine them with the EnKF mentioned above to implement the simultaneous generalization and probabilistic fusion of robot motion trajectories in dual space. The pseudocode of integrating dual-space trajectories, LMO, and phase estimation is presented in Algorithm 1.

---

**Algorithm 1:** Dual-space fusion algorithm for real-time pHRI task.

---

- Input:** (1)  $W = [w_1^\top, \dots, w_N^\top] \in \mathbb{R}^{B \times N}$ : set of  $B$  basis weights corresponding to  $N$  demonstrations, where  $w = [w^H, w^{R_t}, w^{R_j}]$ , which is a combination of the weights of human and robot dual-space trajectories;  
 (2)  $\mathcal{L} = [\ell_1^\top, \dots, \ell_N^\top]$ : LMO parameters obtained from  $N$  demonstrations;  
 (3)  $l = [\frac{1}{T_1}, \dots, \frac{1}{T_N}] \in \mathbb{R}^{1 \times N}$ : reciprocal lengths of demonstrations;  
 (4)  $y_t \in \mathbb{R}^{D \times 1}$ : the trajectory of the observation at time  $t$ .

**Output:**  $\hat{y}_t \in \mathbb{R}^{D \times 1}$ : the inferred trajectory at time  $t$ .

1 Create the initial ensemble (state)  $X_0$ :

$$X_0^j = [0, \delta^j, \ell^j, w^j], \quad 1 \leq j \leq E, i \in [1, N], w^j = w_i, \ell^j = \ell_i, \Phi^j = \frac{1}{T_i}.$$

**while** the interaction is in process **do**

2 For time step  $t$ , propagate the ensemble forward in time as in Equation (22):

$$x_{t|t-1}^j = Gx_{t-1|t-1}^j + \mathcal{N}(0, Q_t), \quad 1 \leq j \leq E.$$

3 If a measurement  $y_t$  is available, perform the measurement update step from Equation (29):

$$X_{t|t} = X_{t|t-1} + K_t(Y_t - H_t X_{t|t-1}).$$

4 Extract the estimated state and uncertainty from the ensemble:

$$\mu_{t|t} = \frac{1}{E} \sum_{j=1}^E x_{t|t-1}^j, \quad \Sigma_{t|t} = \frac{1}{E-1} A_t A_t^\top. \tag{31}$$

5 Compute the generalized trajectories of task space and joint space:

$$\zeta_t^{R_t} = h_{R_t}(\mu_{t|t}), \quad \zeta_t^{R_j} = h_{R_j}(\mu_{t|t}). \tag{32}$$

6 Trajectory point mapping with LMO (Equation (8)):

$$\zeta_t^{R_j^\circ} = \mathcal{L}_t \zeta_t^{R_j}. \tag{33}$$

7 Trajectory point probability fusion and output the controlled trajectory point for the robot:

$$\hat{y}_t = \hat{y}_t^{R_j} = \frac{\zeta_t^{R_j} \Sigma_t^{R_j} + \zeta_t^{R_j^\circ} \Sigma_t^{R_j^\circ}}{\Sigma_t^{R_j} + \Sigma_t^{R_j^\circ}}. \tag{34}$$

8 **end**

9 return  $\hat{y}_t$

---

## 6. Experiments

To evaluate the proposed method, we conducted five experiments that cover different aspects of the problem:

- (1) Handwriting experiment in Section 6.1.1, where we simulated the dual-space features-learning capability, and mapping capability of LMO between different spaces by separating the X, Y, and Z axes of coordinates and treating them as three distinct spaces.
- (2) human-robot-following experiment in Section 6.1.2, where we tested the ability of the probabilistic fusion method to learn both task space and joint space features. We also compared the trajectory error of different methods quantitatively.
- (3) Performance of LMO with different orders in Section 6.2, where we quantified the effect of the Taylor expansion order on the linearization performance of LMO.
- (4) Performance of phase estimation with different filters in Section 6.3, where we evaluated the error and computation time of phase estimation using different filtering techniques.
- (5) Experiment with higher real-time requirements, where we verified that the proposed method can meet the real-time constraints of the task and respond to the fusion of dual-space features.

### 6.1. Learning of Dual-Space Feature

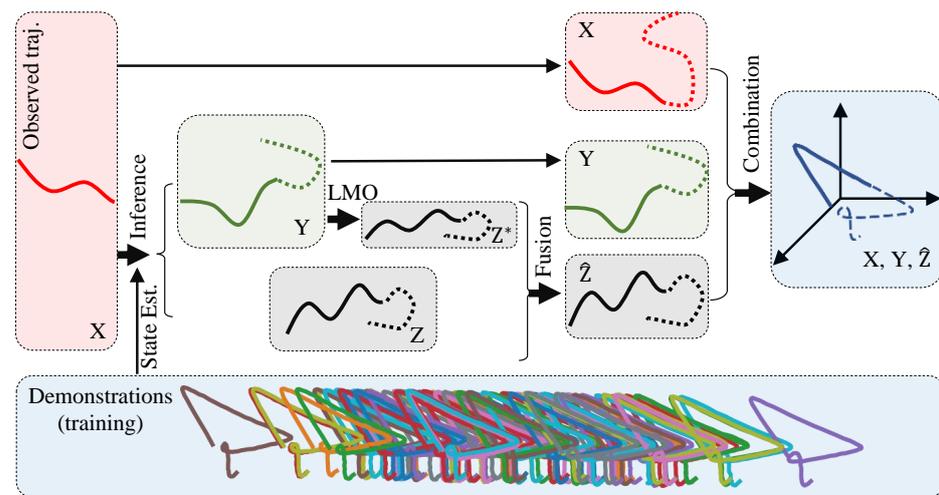
In this part, we will demonstrate the capability of the learning task space feature and joint space feature of the proposed method, two experiments are included in this subsection: (1) a handwriting experiment and (2) a human-robot-following experiment.

#### 6.1.1. Handwriting Experiment

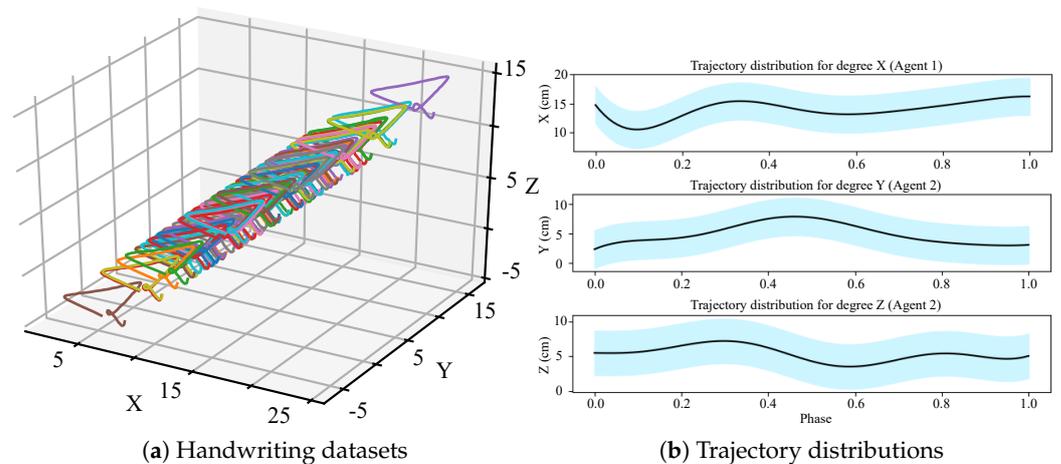
The illustration of this experiment is shown in Figure 2. We generated a handwriting dataset with 100 trajectories based on an opensource one [30], which are visualized in Figure 3a and their distributions are shown in Figure 3b. (The original dataset is two-dimensional (2D). We introduced a random variation in the Z-axis to create a three-dimensional (3D) version of the dataset.) Because this is a simulation experiment, the actual kinesthetic teaching (the leftmost of the top row in Figure 1) and the actual observation process (the leftmost of the bottom row in Figure 1) are no longer needed in this experiment. Instead, the trajectories provided by the dataset are used directly. We abstract an observed agent and a controlled agent, where the X-axis trajectories represent the observed agent, and the Y-axis and Z-axis trajectories represent the controlled agent. That is, the Y-axis and Z-axis trajectories can be regarded as two spatial features of the controlled agent. We think that there is a strong correlation between the different dimensions, and their relationship is constructed with our probabilistic fusion model mentioned in Section 3.

Then, the trajectories of the controlled agent (Y- and Z-axis) are inferred according to the observation (X-axis). We also observe that the two spaces (Y- and Z-axis) of the controlled agent are clearly correlated, even though they have no kinematic relationship. So that we use Equation (8) to estimate the parameters of LMO that capture the correlation between these two different spaces, the illustration of this experiment is shown in Figure 2.

We set the number of basis function components in each dimension to 9. The inference results are shown in Figure 4. We can see that a 3D data trajectory is generated based on the training dataset, which has not been observed before. Furthermore, the trained model infers the remaining motion trajectories of the controlled agents in two spaces (Y and Z) based on the observed partial motion trajectories of the observed agent. This validates that the proposed method can learn and infer different space features and high-dimensional data trajectories that do not have kinematic relationships.



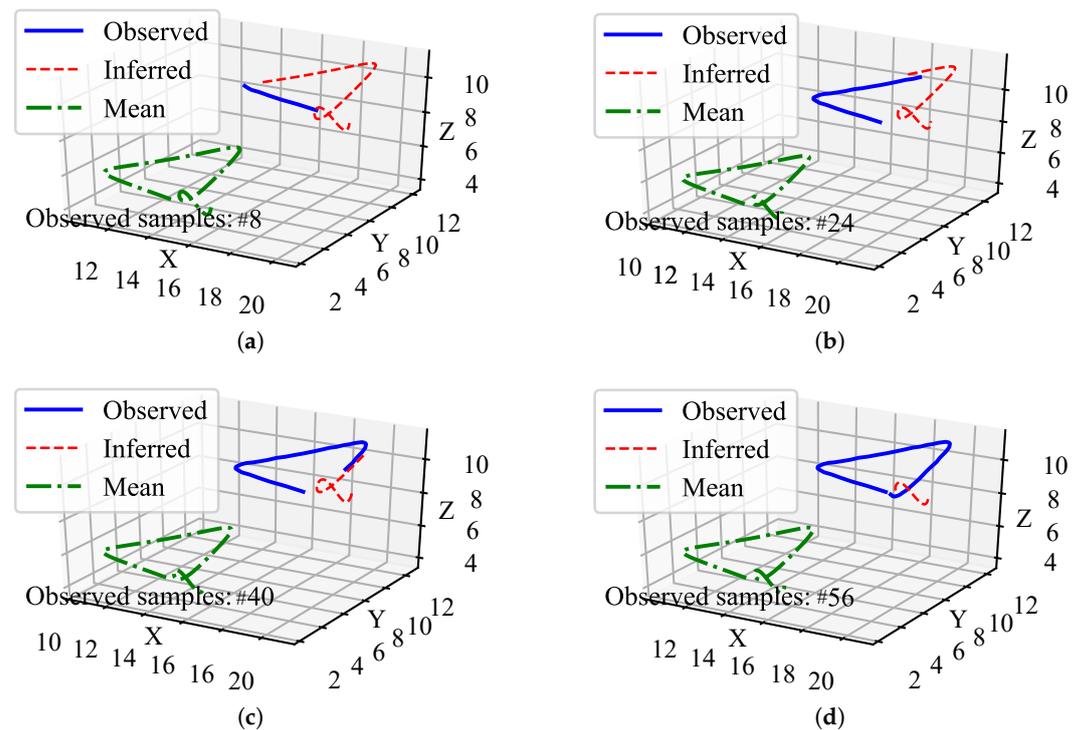
**Figure 2.** Illustration of the handwriting experiment. We consider the X-, Y-, and Z-axis of the trajectories’ coordinates as three distinct spaces, corresponding to different aspects of the human-robot interaction. The X-space represents the task space trajectory of the human or object, the Y-space represents the task space trajectory of the robot, and the Z-space represents the joint space trajectory of the robot, as shown in Figure 1. We train our model with trajectories from all three spaces as input. After training, we can infer a new Y-space trajectory and a new Z-space trajectory by observing a partial X-space trajectory. We can also map the new Y-space trajectory to another Z-space trajectory ( $Z^*$ ) using LMO. The final Z-space trajectory ( $\hat{Z}$ ) is obtained by fusing  $Z$  and  $Z^*$ . Finally, we combine the X, Y, and  $\hat{Z}$  trajectories into a 3D trajectory and compare it with the demonstration.



**Figure 3.** The handwriting trajectories and inference. (a) The 100 3D trajectories of handwriting, that is those curves with different colors in the figure; (b) the trajectory distributions of X (top), Y (middle), and Z (bottom) axes.

6.1.2. human-Robot-following Experiment

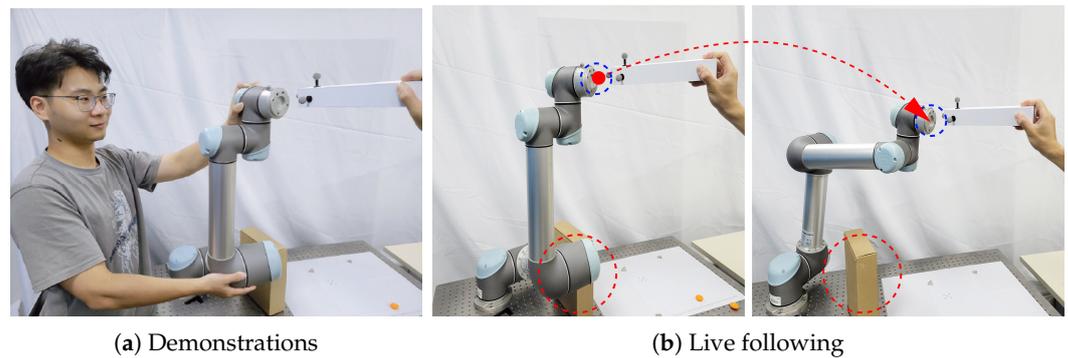
We also performed a real human-robot-following task which is shown in Figure 5. We collected a total of 60 demonstrations, where 45 were used for training and the remaining 15 for testing. The trajectories were formatted and divided into three parts: the trajectories of the human-held object (observed object), the end-effector trajectories of the UR5 manipulator (task space trajectories), and the trajectories of each joint angle of the UR5 manipulator (joint space trajectories). The three types of trajectories are modeled by fitting Gaussian basis functions of different numbers and scales (the scale is defined as the width of the basis [7]). In interactive inference, our algorithm performs real-time observation of the target object, simultaneously inferring the trajectories in both spaces and generating fused trajectories to control the robot to satisfy both task space and joint space requirements.



**Figure 4.** Examples of the handwriting trajectories inference. Four examples (a–d) are visualized with different observed samples (#8, #24, #40, and #56). The green dashed lines depict the mean of data trajectories. The blue solid lines depict the observed data trajectories for testing. Furthermore, the red dotted lines depict the inferred data trajectories. Since we only observed the trajectories in the X-axis, coordinates in one direction cannot be visualized in a three-dimensional coordinate system, the observed trajectories shown in the figures are combined with both the X-, Y-, and Z-axis of the observed sample, but the trajectories in Y- and Z-axis are not observed in the whole inference process.

The top row of Figure 6 presents the process of observing human movement and inferring collision-free trajectories in both joint space and task space with our method. It can be found that the robot can perform the following task with human guidance and its joints go well around the box, i.e., the robot can satisfy the common constraints of both spaces and the dual-space trajectories obtained by synchronous inference are feasible and valid.

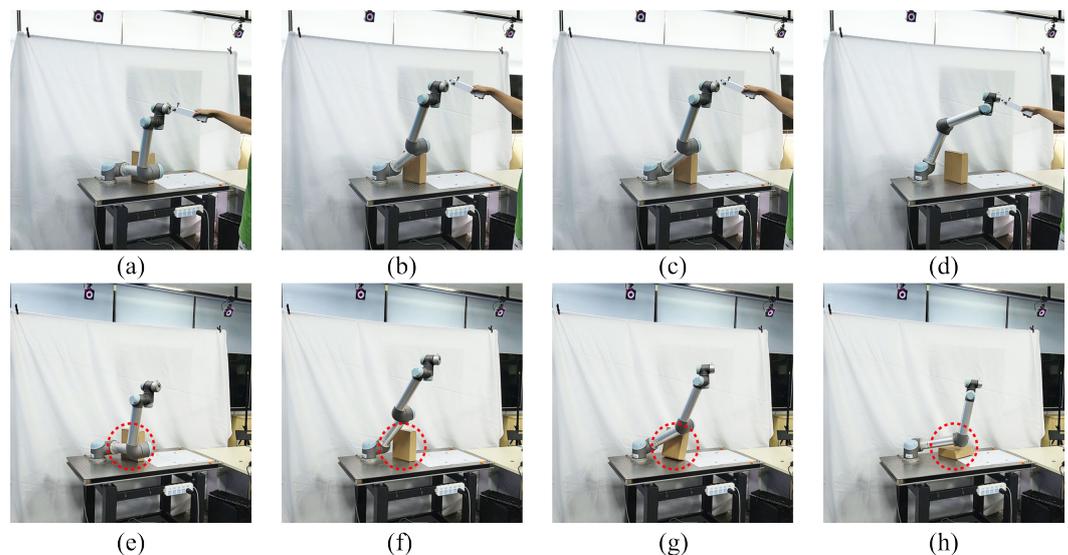
As a comparison, we use the same training set (excluding robot joint space trajectories) and Gaussian basis functions to model only task space inference in the same way as the IP framework, the parameters are shown in Table 2. In this case, the IP is only inferred for the task space control trajectory of the robot, its results are shown in the bottom row of Figure 6. It can be seen that the robot follows the target object when only task space inference is performed. For the sake of controlling variables, when performing the experiment of IP, we replay the observed trajectories of the object recorded at the experiment of our method; that is, the data played back here are the same as the trajectory data of the human-held object in the top row of Figure 6. It can be seen that considering only the task space requirements (IP), the robot is not guaranteed to fulfill the joint space constraints (the robot knocks over the box). On the contrary, motion generated with our method can not only accurately achieves the goal, but also learns the joint space features effectively and ensures the collision-free requirements. This verifies the effectiveness of the proposed method in learning dual-space features.



**Figure 5.** Illustration of the human-robot-following task. (a): The robot was demonstrated to follow the target object and avoid colliding with the obstacle. (b): during the live interaction, a new guidance was performed by the user (the red arrow), and the robot can generate a collision-free (the red circle) trajectory with the proposed method.

**Table 2.** Basis function parameters of trajectories for the human-robot-following task.

Trajectory	Number of Basis Functions	Scale of Basis
Task space trajectory (object)	13	0.01
Task space trajectory (robot)	9	0.01
Joint motion trajectory (robot)	9	0.04



**Figure 6.** Snapshots of the human-robot-following experiment. **Top:** the interaction process of the proposed method. (a–d) represent screenshots of different moments of the process from start to stop, and (e–h) do the same. The end-effector of the UR5 manipulator follows the movement of a human handheld object while fulfilling the joint space requirements. **Bottom:** the interaction process that is derived from using IP to model and infer the trajectory of the task space. The red circle depicts that there is a collision between the robot and the obstacle. To ensure that both methods use the same observed trajectory for inference, we set the observed task trajectory to be identical to ours. This means that, in the live interaction of IP, we replay the trajectory of the observed object that was recorded in the experiment using our proposed method.

We also quantified the error of the inferred trajectories using different methods and spaces. We first modeled the inference of the joint space only with IP. Then, we compared the performance of three inference approaches: task space only (IP), joint space only (IP), and dual-space fusion (ours). We used the mean absolute error (MAE) and root mean squared error (RMSE) metrics to evaluate the trajectories generated by each approach on

the same test set. Table 3 shows the results. For each group of demonstrations in the test set, we observed the target object trajectories (human actions) and generated the robot control trajectories using each of the three approaches in both spaces. We then computed the RMSE and MAE between the robot trajectories of the test set (demo trajectories) and the trajectories generated by each approach in each space at each time step.

Let  $M$  be the number of demonstrations (that is,  $M = 15$ ) in the test set. Assume that there are  $N_i$  interaction classes. (An interaction class of a time step is defined as a tuple of the observed object’s position, end-effector’s position, and robot’s joint angles.) In the  $i$ -th ( $i \in [1, M]$ ) demonstration, let  $\mathbf{p}_j^A$  be robot end-effector’s coordinates in the  $j$ -th interaction class of  $i$ -th demonstration; that is,  $\mathbf{p}_j^A = [x_j^A, y_j^A, z_j^A]$  (A for actual). Furthermore, let  $\mathbf{p}_j^I$  be the corresponding coordinates that are predicted with different methods; that is,  $\mathbf{p}_j^I = [x_j^I, y_j^I, z_j^I]$  (I for inferred). Then, the MAE and RMSE for the task space are defined as:

$$MAE = \frac{1}{M} \sum_{i=1}^M \frac{1}{N_i} \sum_{j=1}^{N_i} (|x_j^I - x_j^A| + |y_j^I - y_j^A| + |z_j^I - z_j^A|), \tag{35}$$

$$RMSE = \frac{1}{M} \sum_{i=1}^M \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} (\|\mathbf{p}_j^I - \mathbf{p}_j^A\|_2^2)}. \tag{36}$$

For the joint space, let  $\theta_j^A$  be the robot’s joint angles in the  $j$ -th interaction class of  $i$ -th demonstration; that is,  $\theta_j^A = [\theta_{j1}^A, \theta_{j2}^A, \dots, \theta_{jK}^A]$ , where  $K$  is the degrees of freedom of the robot, as well as the inferred joint angles  $\theta_j^I = [\theta_{j1}^I, \theta_{j2}^I, \dots, \theta_{jK}^I]$ . Then, the MAE and RMSE for the joint space are defined as:

$$MAE = \frac{1}{M} \sum_{i=1}^M \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{1}{K} \sum_{k=1}^K |\theta_{jk}^I - \theta_{jk}^A|, \tag{37}$$

$$RMSE = \frac{1}{M} \sum_{i=1}^M \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} (\|\theta_j^I - \theta_j^A\|_2^2)}. \tag{38}$$

As shown in Table 3, the proposed method achieves the best performance in all comparison metrics. The joint space trajectories have large deviations (RMSE:  $3.4 \times 10^{-1}$  rad) when only the task space trajectories are inferred, which is consistent with the result shown in Figure 6. When only the joint space is inferred, the robot can satisfy the joint space constraint (RMSE:  $8.5 \times 10^{-2}$  m), but its end-effector trajectory accuracy (MAE:  $2.2 \times 10^{-2}$  m) is worse than that of our dual-space fusion method (MAE:  $8.4 \times 10^{-3}$  m). Moreover, UR5 is a 6 DoF manipulator, so its end-effector position accuracy is less influenced by joint generalization than that of a redundant robot.

**Table 3.** Comparison of trajectory errors among three inference methods.

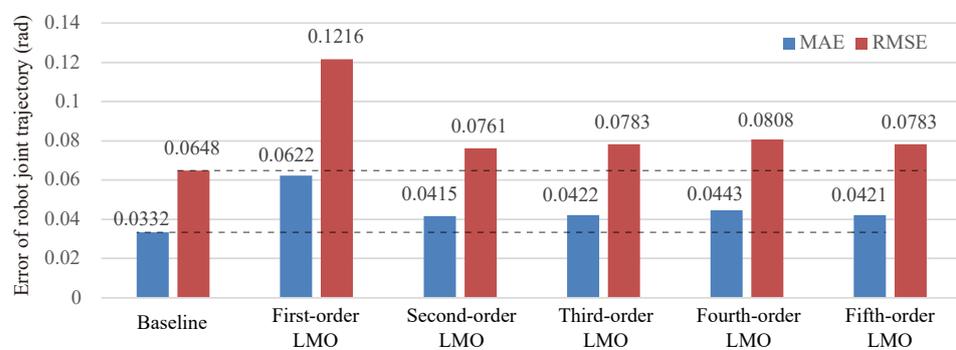
	Task Space		Joint Space	
	RMSE (m)	MAE (m)	RMSE (rad)	MAE (rad)
Task space only (IP)	$4.3 \times 10^{-2}$	$2.9 \times 10^{-2}$	$3.4 \times 10^{-1}$	$9.9 \times 10^{-2}$
Joint space only (IP)	$3.5 \times 10^{-2}$	$2.2 \times 10^{-2}$	$1.3 \times 10^{-1}$	$6.9 \times 10^{-2}$
Dual-space fusion (ours)	$1.2 \times 10^{-2}$	$8.4 \times 10^{-3}$	$8.5 \times 10^{-2}$	$4.0 \times 10^{-2}$

As demonstrated by the above experiments, our dual-space probabilistic fusion approach can handle HRI tasks with joint constraints in both task space and joint space and enhance the accuracy of single-space inference in either task space or joint space.

### 6.2. Performance of LMO with Different Orders

It is impossible to expand Equation (10) to an infinite series because the more that the series are expanded, the more parameters are involved, and the higher the computational time consumption. Therefore, we conducted this experiment to investigate the effect of the order of LMO on inference accuracy, and finally determine how many series should be expanded.

The same dataset as the human-robot-following task in Section 6.1.2 is used here. We compared our method with different orders (first to fifth order) of LMO to our previous work (baseline) [13], in which the robot kinematics are used. In our method, we approximate the kinematics mapping from the task space to the joint space by LMO. To ensure a fair comparison, the same number of basis function components and the same scaling factor in basis functions are applied for all methods. Furthermore, the EnKF is replaced with EKF which is the same as the baseline. The MAE and the RMSE are calculated to measure the error between the inferred trajectories in robot joint space and the ground truth, and the quantitative results are summarized in in Figure 7. The results show that the MAE decreased by 33% and the RMSE decreased by 37% when LMO is switched from the first order to the second order. However, increasing the order of LMO further does not lead to significant reductions in MAE and RMSE. This is because the Taylor expansion is a method of approximating a function by an infinite series. The more terms in the series, the closer the approximation is to the true value of the function. However, as the number of terms increases, the convergence rate of the approximation slows down. The MAE of second-order LMO is 20% higher than that of the baseline, and the RMSE is 15% higher than that of the baseline. From the above relative comparison, it can be seen that the inference accuracy of LMO is somewhat lower than that of the baseline (using the robot's kinematics model directly). However, from the absolute values, the second-order LMO is very close to the baseline, and both MAE and RMSE maintain a very small value, which can meet the requirements of most HRI tasks.



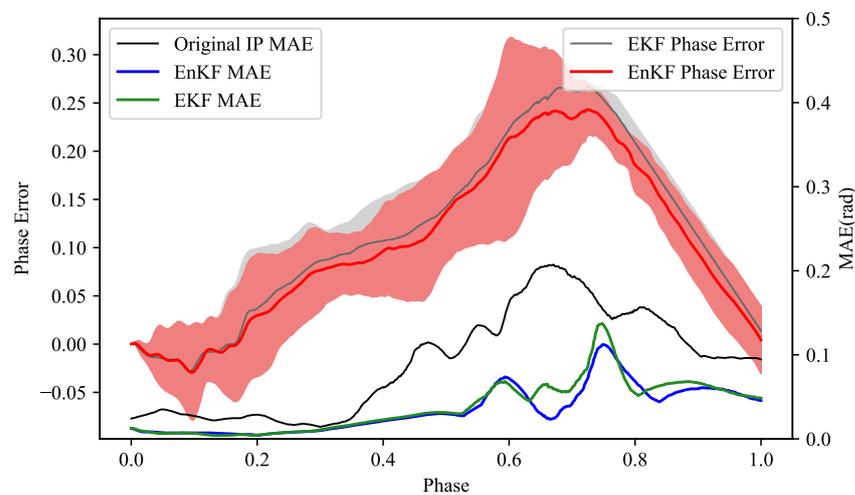
**Figure 7.** Comparisons of inference accuracy with different orders of LMO. The baseline is our previous work [13] in which the robot kinematics are used.

### 6.3. Performance of Phase Estimation with Different Filters

This experiment consists of two parts: the first part is to compare and analyze the inference accuracy and phase estimation accuracy of different methods (IP, EKF, and EnKF), and the second part is to compare the inference computation with using different filters (EKF and EnKF). In each group of comparison experiments, training data, samples, and parameter settings are the same.

Figure 8 compares and evaluates the phase estimation error and trajectory accuracy of EKF, EnKF, and the original IP. The gray line in the figure shows the mean phase error of the entire interaction process obtained by EKF for the testing set, and the gray area shows the ensemble range formed by different testing samples. Similarly, the red line and red area show the mean phase error and ensemble range obtained by EnKF for the same testing set. The right vertical axis in the figure indicates the phase estimation error obtained by different filters. Comparing EKF, EnKF, and the original IP, we obtain three mean error lines at the

bottom of Figure 8. These three error lines and Table 4 reveal that the dual-space feature inference of the Kalman Filter framework is more accurate than the original IP, which is consistent with the conclusion of human-robot-following in Section 6.1.2. Moreover, EnKF achieves better results than EKF, but their results are overall similar, which is also reflected in their phase estimation comparison. This is because they share a similar modeling idea for phase estimation, so in the interaction process, for the same training set, their inference mean lines and distribution situations are very close. Due to the introduction of phase parameters and LMO, the dimension of the state vector has increased significantly. If we use EKF, it involves a large-scale operation process on the covariance matrix, and its computational complexity is  $\mathcal{O}(N^3)$ , where  $N$  denotes the dimension of the state vector. For EnKF (this paper), we use a set of dimension  $E$  to represent the state matrix, which is essentially a kind of dimension reduction, and its computational complexity is reduced to  $\mathcal{O}(E^2N)$ . The selected set size ( $E$ ) is generally much smaller than the state dimension ( $N$ ), which will contribute to improving the performance of computation.



**Figure 8.** Comparison of the inferred performance of different methods.

**Table 4.** Comparison of the inference accuracy among the three methods.

Metrics (rad)	Original IP	EKF	EnKF
MAE	$9.08 \times 10^{-2}$	$3.82 \times 10^{-2}$	$3.44 \times 10^{-2}$

The dataset recorded in the last experiment is input into the algorithms in the form of data playback, simulating an interaction scenario, and then recording the time required for each iterative inference. The computational cost is obtained, visualized, and analyzed in Figure 9 and Table 5.

As shown in Figure 9, the iteration speed of EnKF is generally higher than that of EKF, and it can also be found that the time required for EKF is unstable and fluctuates greatly (the gray line in the figure has obvious sawtooth fluctuations). On the contrary, EnKF has a relatively stable inference time at each interaction moment, and its overall variance is small. As can be seen from Table 5, compared with EKF, the overall inference speed of EnKF is increased by 54.87%. Combined with the results of the previous experiment, it can be concluded that the EnKF spatiotemporal synchronous inference method adopted in this paper achieves higher improvement in computational performance while ensuring inference accuracy, and is more suitable for real-time inference in HRI.

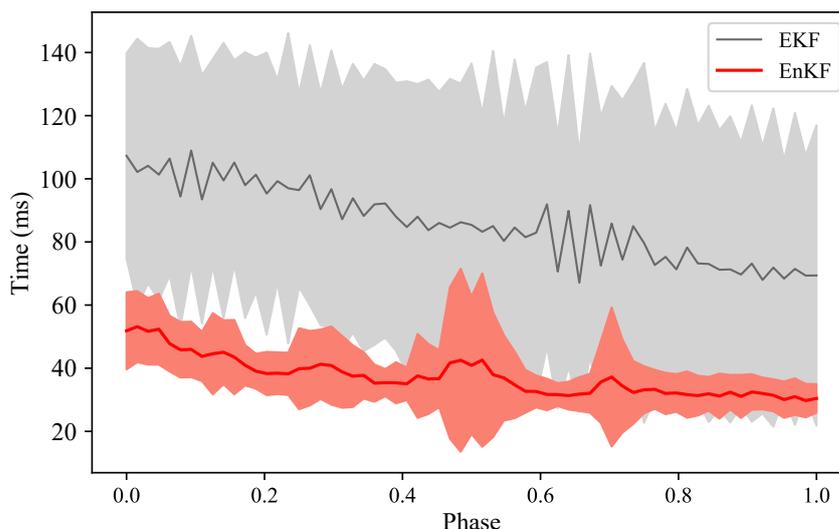


Figure 9. Comparison of the computational performance between the two methods.

Table 5. Comparison of the mean computational costs between EKF and EnKF.

Metrics	EKF	EnKF
mean cost time (ms/epoch)	83.17	37.39

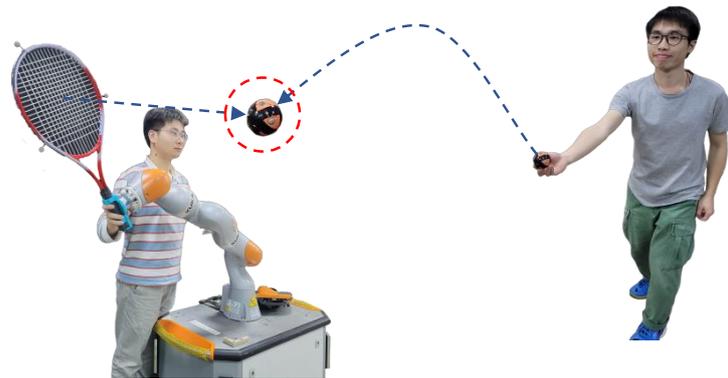
#### 6.4. An Experiment with Higher Real-Time Requirements

In order to demonstrate the real-time inference ability of the proposed framework for HRI tasks, we conducted a human-robot collaborative ball-hitting experiment, as shown in Figures 1 and 10. Two human participants were involved in the data collection process. One participant threw a ball from different distances and angles, whereas the other participant demonstrated the hitting motion to a KUKA iiwa7 robot with a racket attached to its end-effector via kinesthetic teaching. Trajectories of the ball (task space) were recorded by an Optitrack motion capture system at 120 Hz. The robot’s joint motion trajectories (joint space) and its end-effector pose trajectories (task space) were recorded via ROS. Furthermore, a total of 140 training samples were collected. The fitting parameters of each type of trajectory were determined by BIC, and the specific Gaussian radial basis function parameters are shown in Table 6.

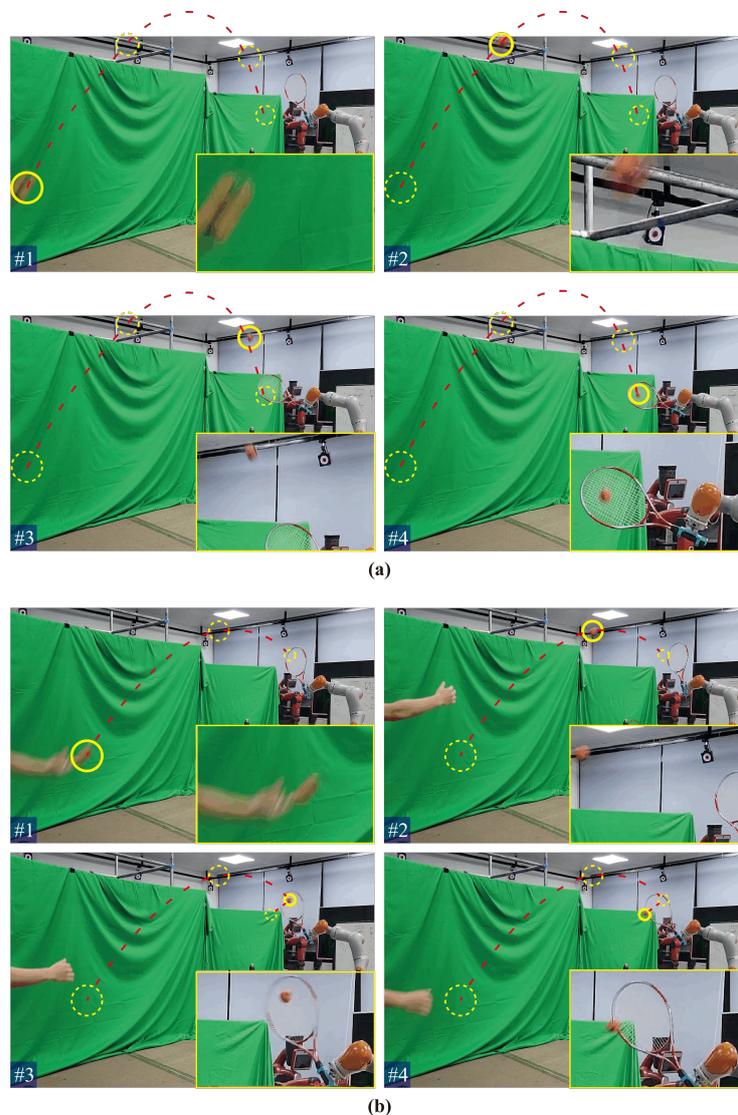
The snapshots of the live human-robot ball-hitting experiment are shown in Figure 11. In the live interaction, the trained model simultaneously infers the robot motion trajectories in the robot joint space and task space according to the observed ball motion trajectories and controls the robot to complete the hitting. If the inference accuracy and real-time performance are insufficient, it will cause the robot to miss the hitting point. The four pictures in Figure 11a together show that our inference algorithm observes the ball’s trajectory and generates the response trajectory for the robot. Furthermore, the robot synchronizes its movement with the ball’s parabolic motion and hits the ball (task space) at the predicted point using a similar configuration (joint space) as the demonstrations. Similarly, Figure 11b shows a higher landing point for the ball trajectory, and the robot makes a downward hitting action with a racket. The two interaction processes demonstrate our online adaptive ability and verify that our inference performance can perform real-time pHRI tasks. The video of the human-robot-following experiment and ball-hitting experiment is Supplementary Materials.

Table 6. Basis function parameters of modeling for ball-hitting experiment.

Trajectory	Number of Basis Functions	Scale of Basis
Ball position trajectory	13	0.01
Robot pose trajectory	9	0.01
Robot joint motion trajectory	9	0.04



**Figure 10.** Illustration for the human-robot ball-hitting experiment. The blue dotted line depicts the motion trajectories of the ball and the racket, and the red circle depicts hitting point. Several reflective markers are attached to the ball and the racket to capture their motion.



**Figure 11.** Snapshots of the live human-robot ball-hitting task. We present two groups (a,b) of hitting experiments in the figure, each containing four sub-figures (#1, #2, #3, and #4). The red dotted lines represent the ball motion trajectories, the yellow solid circles represent the current position of the ball, and the yellow dashed circles represent the position of the ball at other moments. In addition, a partially enlarged view is placed at the right bottom corner of each sub-figure, which shows the current state of the ball, e.g., the one in #4 is the scene of the ball being hit by the racket.

## 7. Conclusions

This paper proposes a probabilistic dual-space (task space and joint space) feature fusion for real-time physical human-robot interaction. We conducted several qualitative and quantitative experiments and analyses to validate our effectiveness. Notably, we achieved:

- A pHRI method that considers the dual-space features of the robot. The dual-space trajectory inference fusion method proposed in this paper addresses the limitation of the original IP, which can only perform single-space generalization. By applying our method with two datasets of a UR5 and a KUKA iiwa7 robot, we reduce the trajectory inference error of the proposed method by more than 33% in both spaces. In addition, we demonstrate the applicability of the proposed method to different robots by verifying it with two datasets of robots with different DoFs. Moreover, the inferred results satisfy both task space and joint space constraints and can meet real-time inference interaction requirements.
- A dual-space linear mapping method that does not depend on the robot model. We propose a method to learn and infer the linear mapping relationship between different spaces from demonstrations. Through the three-dimensional handwriting trajectory experiment, we verify the feasibility of integrating the linear mapping operator into the dual-space synchronous inference framework. With the UR5 dataset experiment, we examine the influence of the expansion order of the linear mapping operator on the trajectory inference accuracy, and we show that increasing the order of the mapping operator within a certain range can improve the inference trajectory accuracy. In the experiment, the trajectory MAE obtained by the second-order LMO is close to that obtained by the kinematic mapping.
- We integrate phase estimation into the Kalman Filter framework to realize a method for spatiotemporal synchronous inference, which reduces the inference hierarchy. Based on the idea of ensemble sampling, we use ensemble Kalman Filter inference to solve the high-dimensional inference problem caused by dual-space inference. The experiments show that our improvements maintain the accuracy superiority of dual-space feature learning, and that the overall computational efficiency is 54.87% higher than EKF.

Our method learns both the task space and joint space features of the demonstration, enabling it to handle scenarios that require both target position and configuration, especially for real-time tasks. As humanoid robots and related technologies become more integrated into people's daily lives, our method can be applied to service robots to generate motions that satisfy simultaneous constraints on multiple robot joint spaces and task spaces, such as robot anthropomorphic writing, dancing, sign language imitation and communication, greeting, etc. These tasks not only depend on the position of the robot's hand (end-effector), but also on whether the robot's configuration (shape) is human-like. In the future, we will explore using our method to learn more feature fusion, such as speed, acceleration, force, etc., so that the generated motion exhibits more human-like characteristics, enhancing its applicability in practical HRI and improving the interaction efficiency and human acceptance of robots.

**Supplementary Materials:** The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/biomimetics8060497/s1>.

**Author Contributions:** Conceptualization, X.C. and Y.L.; methodology, Y.L. and J.W.; software, Y.L. and J.W.; validation, Y.L. and J.W.; formal analysis, Y.L. and X.C.; investigation, X.C. and J.W.; resources, Y.L. and J.W.; data curation, X.C. and Y.L.; writing—original draft preparation, Y.L. and J.W.; writing—review and editing, Y.L., J.W. and Y.G.; visualization, Y.L. and J.W.; supervision, Y.G.; project administration, Y.G.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work in this paper is in part supported by the Guangdong Yangfan Program for Innovative and Entrepreneurial Teams (Grant No. 2017YT05G026), the Natural Science Foundation of China (Grant No. 51975126), and the Key Research and Development Program of Guangdong Province (Grant No. 2019B090915001).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data related to this paper will be made available upon request. The access to the data is subject to the data protection policy of the Biomimetic and Intelligent Robotics Lab (BIRL) and Guangdong University of Technology (GDUT), China.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Haddadin, S.; Croft, E. Physical human-Robot Interaction. In *Springer Handbook of Robotics*; Springer International Publishing: Cham, Switzerland, 2016; pp. 1835–1874. [\[CrossRef\]](#)
2. Wu, H.; Yan, W.; Xu, Z.; Cheng, T.; Zhou, X. A framework of robot skill learning from complex and long-horizon tasks. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 3628–3638. [\[CrossRef\]](#)
3. Wu, H.; Yan, W.; Xu, Z.; Zhou, X. A Framework of Improving Human Demonstration Efficiency for Goal-Directed Robot Skill Learning. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *14*, 1743–1754. [\[CrossRef\]](#)
4. Asfour, T.; Azad, P.; Gyarfas, F.; Dillmann, R. Imitation learning of dual-arm manipulation tasks in humanoid robots. *Int. J. Humanoid Robot.* **2008**, *5*, 183–202. [\[CrossRef\]](#)
5. Qin, F.; Xu, D.; Zhang, D.; Li, Y. Robotic skill learning for precision assembly with microscopic vision and force feedback. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 1117–1128. [\[CrossRef\]](#)
6. Ijspeert, A.J.; Nakanishi, J.; Hoffmann, H.; Pastor, P.; Schaal, S. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Comput.* **2013**, *25*, 328–373. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Paraschos, A.; Daniel, C.; Peters, J.R.; Neumann, G. Probabilistic movement primitives. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–6 December 2013; pp. 2616–2624. [\[CrossRef\]](#)
8. Huang, Y.; Rozo, L.; Silvério, J.; Caldwell, D.G. Kernelized movement primitives. *Int. J. Robot. Res.* **2019**, *38*, 833–852. [\[CrossRef\]](#)
9. Calinon, S.; Billard, A. Statistical learning by imitation of competing constraints in joint space and task space. *Adv. Robot.* **2009**, *23*, 2059–2076. [\[CrossRef\]](#)
10. Stilman, M. Global manipulation planning in robot joint space with task constraints. *IEEE Trans. Robot.* **2010**, *26*, 576–584. [\[CrossRef\]](#)
11. Chen, Y.; Luo, X.; Han, B.; Luo, Q.; Qiao, L. Model predictive control with integral compensation for motion control of robot manipulator in joint and task spaces. *IEEE Access* **2020**, *8*, 107063–107075. [\[CrossRef\]](#)
12. Wang, S.; Murphy, K.; Kenney, D.; Ramos, J. A comparison between joint space and task space mappings for dynamic teleoperation of an anthropomorphic robotic arm in reaction tests. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 2846–2852. [\[CrossRef\]](#)
13. Chen, X.; Li, Y.; Guan, Y.; Shi, W.; Wu, J. Probabilistic Fusion in Task Space and Joint Space for Human-Robot Interaction. In Proceedings of the 2022 IEEE International Conference on Robotics and Biomimetics (ROBIO), Xishuangbanna, China, 5–9 December 2022; pp. 415–420. [\[CrossRef\]](#)
14. Amor, H.B.; Neumann, G.; Kamthe, S.; Kroemer, O.; Peters, J. Interaction primitives for human-robot cooperation tasks. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 2831–2837. [\[CrossRef\]](#)
15. Cui, Y.; Poon, J.; Miro, J.V.; Yamazaki, K.; Sugimoto, K.; Matsubara, T. Environment-adaptive interaction primitives through visual context for human-robot motor skill learning. *Auton. Robot.* **2019**, *43*, 1225–1240. [\[CrossRef\]](#)
16. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent advances in robot learning from demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [\[CrossRef\]](#)
17. Silvério, J.; Huang, Y.; Rozo, L.; Caldwell, D.G. An uncertainty-aware minimal intervention control strategy learned from demonstrations. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 6065–6071. [\[CrossRef\]](#)
18. Vogt, D.; Lorenz, B.; Grehl, S.; Jung, B. Behavior generation for interactive virtual humans using context-dependent interaction meshes and automated constraint extraction. *Comput. Animat. Virtual Worlds* **2015**, *26*, 227–235. [\[CrossRef\]](#)
19. Vogt, D.; Stepputtis, S.; Grehl, S.; Jung, B.; Amor, H.B. A system for learning continuous human-robot interactions from human-human demonstrations. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 2882–2889. [\[CrossRef\]](#)
20. Wang, Z.; Mülling, K.; Deisenroth, M.P.; Ben Amor, H.; Vogt, D.; Schölkopf, B.; Peters, J. Probabilistic movement modeling for intention inference in human-robot interaction. *Int. J. Robot. Res.* **2013**, *32*, 841–858. [\[CrossRef\]](#)

21. Ikemoto, S.; Amor, H.B.; Minato, T.; Jung, B.; Ishiguro, H. Physical human-robot interaction: Mutual learning and adaptation. *IEEE Robot. Autom. Mag.* **2012**, *19*, 24–35. [[CrossRef](#)]
22. Calinon, S.; Billard, A. A probabilistic programming by demonstration framework handling constraints in joint space and task space. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 367–372. [[CrossRef](#)]
23. Stilman, M. Task constrained motion planning in robot joint space. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 3074–3081. [[CrossRef](#)]
24. Chen, L.; Wu, H.; Duan, S.; Guan, Y.; Rojas, J. Learning human-robot collaboration insights through the integration of muscle activity in interaction motion models. In Proceedings of the 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, UK, 15–17 November 2017; pp. 491–496. [[CrossRef](#)]
25. Li, Y.; Wu, J.; Chen, X.; Guan, Y.; Zhu, H. TMG: A topology-based motion generalization method with spatial relationship preservation. *Robot. Auton. Syst.* **2023**, *166*, 104445. [[CrossRef](#)]
26. Schneider, M.; Ertel, W. Robot learning by demonstration with local gaussian process regression. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 255–260. [[CrossRef](#)]
27. Li, Z.; Hong, X.; Hao, K.; Chen, L.; Huang, B. Gaussian process regression with heteroscedastic noises—A machine-learning predictive variance approach. *Chem. Eng. Res. Des.* **2020**, *157*, 162–173. [[CrossRef](#)]
28. Zhang, Q.H.; Ni, Y.Q. Improved most likely heteroscedastic Gaussian process regression via Bayesian residual moment estimator. *IEEE Trans. Signal Process.* **2020**, *68*, 3450–3460. [[CrossRef](#)]
29. Huang, Y.; Silvério, J.; Rozo, L.; Caldwell, D.G. Hybrid probabilistic trajectory optimization using null-space exploration. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 7226–7232. [[CrossRef](#)]
30. Campbell, J.; Amor, H.B. Bayesian interaction primitives: A slam approach to human-robot interaction. In Proceedings of the Conference on Robot Learning (CoRL), Mountain View, CA, USA, 13–15 November 2017; pp. 379–387.
31. Campbell, J.; Stepputtis, S.; Amor, H.B. Probabilistic Multimodal Modeling for Human-Robot Interaction Tasks. In Proceedings of the 15th Robotics: Science and Systems (RSS), Freiburg, Germany, 22–26 June 2019; MIT Press Journals: Cambridge, MA, USA, 2019. [[CrossRef](#)]
32. Maeda, G.; Ewerton, M.; Neumann, G.; Lioutikov, R.; Peters, J. Phase estimation for fast action recognition and trajectory generation in human-robot collaboration. *Int. J. Robot. Res.* **2017**, *36*, 1579–1594. [[CrossRef](#)]
33. Aczel, B.; Hoekstra, R.; Gelman, A.; Wagenmakers, E.J.; Klugkist, I.G.; Rouder, J.N.; Vandekerckhove, J.; Lee, M.D.; Morey, R.D.; Vanpaemel, W.; et al. Discussion points for Bayesian inference. *Nat. Hum. Behav.* **2020**, *4*, 561–563. [[CrossRef](#)]
34. Seeger, M. Gaussian processes for machine learning. *Int. J. Neural Syst.* **2004**, *14*, 69–106. [[CrossRef](#)]
35. Cui, Z.; Dai, J.; Sun, J.; Li, D.; Wang, L.; Wang, K. Hybrid methods using neural network and Kalman filter for the state of charge estimation of lithium-ion battery. *Math. Probl. Eng.* **2022**, *2022*, 9616124. [[CrossRef](#)]
36. Kim, T.; Park, T.H. Extended Kalman filter (EKF) design for vehicle position tracking using reliability function of radar and lidar. *Sensors* **2020**, *20*, 4126. [[CrossRef](#)] [[PubMed](#)]
37. Van Leeuwen, P.J. A consistent interpretation of the stochastic version of the Ensemble Kalman Filter. *Q. J. R. Meteorol. Soc.* **2020**, *146*, 2815–2825. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.