This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

Fast Decision Making in Human-Machine Interaction through Estimation of Human Reaching Intent

Jasper Van der Auwera, Erwin Aertbeliën, Wilm Decré, *Member, IEEE*, and Herman Bruyninckx

***Abstract*—This paper proposes intent-based decision-making approaches for operator assistance in manipulation tasks, where the intent of the human operator is defined by the target of an ongoing reaching motion. These approaches start from a motion model database, which contains multiple motion models learned using Trajectory parameterized Probabilistic Principal Component Analysis (TraPPCA). Often, such a database is constructed offline using supervised learning. Instead, this paper proposes a semi-supervised and an unsupervised online learning approach, able to encode new motion styles observed during execution. Then, the intended target of an ongoing reaching motion is predicted based on this motion model database, for which three probabilistic intent estimation approaches are proposed: (1) classification intent estimation, (2) prediction intent estimation, and (3) mixture intent estimation. Additionally, a decision-making approach is introduced which determines whether the human operator will deviate from the task's high-level plan based on the target probabilities obtained from intent estimation, such that an early intervention in the human's task execution can be provided. This decision-making approach directly considers the target probabilities in its decision-making process, instead of only considering the most probable target, which makes it more adaptable to different situations and better at dealing with large uncertainties. The proposed methods were validated on the publicly available HA4M dataset.**

***Index Terms*—Decision making, intent estimation, Learning from Demonstration (LfD), online learning, operator assistance**

I. Introduction

HUMAN-MACHINE INTERACTION (HMI) aims to create mechanisms for intuitive collaboration and natural communication between humans and different types of machines, such as robotic devices (e.g., a robot arm [1] or an exoskeleton [2]) in human-robot collaboration (HRC) and assistive devices (e.g., operator assistance systems [3]) in human task monitoring. However, to obtain efficient, intuitive interactions, the machine should be able to anticipate the behavior of the human operator [4], similar to interactions between humans [5], which illustrates the need for prediction and intent estimation of ongoing human motions.

This paper focuses on intent estimation and decision making for operator assistance in manipulation tasks, where the *intent* of the human operator is defined by the target of an ongoing reaching motion. This target is either one of the predefined targets of the task, determined by their location, or is outside the scope of the task. Regardless, the target is assumed to be constant throughout the reaching motion.

This work aims to improve operator assistance in manipulation tasks by improving probabilistic intent estimation and its integration with high-level decision making. Here, the predictions of the intent are represented by a probability distribution over the potential targets. The specific feedback (e.g., lights, projections, visualization on screen, augmented reality) is outside the scope of this paper and is not further discussed. The type of application considered in this work is illustrated in Fig. 1, depicting the high-level plan, the workspace setup, and the reaching motions for container picking in the HA4M dataset [6], which will be further elaborated in section IV.

Crucial for predicting human motions and estimating human intent are mathematical representations of human motions and their allowable variations. To obtain these mathematical representations, this work focuses on imitation learning approaches with two main characteristics: (1) an *explicit progress dependency* and (2) the ability to construct probabilistic motion models based on a *small number of demonstrations*. The explicit progress dependency facilitates reasoning about the future, without further integration or simulation, which is beneficial when predicting human motions and estimating human intent. However, it also introduces the additional challenge of estimating the progress along a partially observed trajectory. The small number of demonstrations is considered important to enable quick deployment. In this work, Trajectory parameterized Probabilistic Principal Component Analysis (TraPPCA) [7] was selected as the imitation learning approach. However, the main message of this work does not depend on this particular motion modeling approach.

The considered imitation learning approaches have a limited number of parameters, limiting the variations that can be represented by a single motion model. Therefore, a motion model database will be constructed, containing separate motion models for different motion types. Motion model databases are generally learned using supervised learning methods, which require an offline training phase and manually-labelled training data. However, human task performance can change over time (e.g., due to fatigue) or different performances can be observed for different operators. In such cases, models trained offline will no longer properly represent the motions performed by the human operator, resulting in less accurate predictions. Therefore, in section III-D, this paper introduces novel semi-supervised and unsupervised online learning algorithms, aiming to encode variations that occur during execution into the

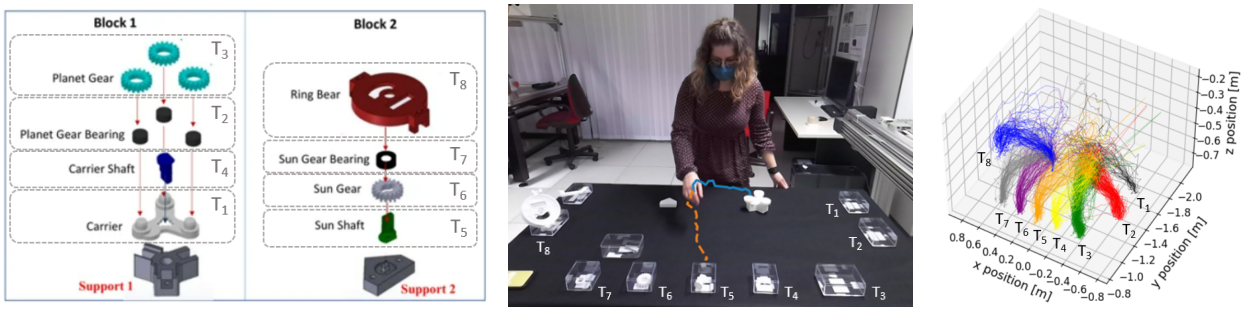
Manuscript received ... This work was supported by Flanders Make (Flanders Make is the Flemish Strategic Research Center for the manufacturing industry), through the Flanders Make SBO project AUTOCRAFT.

Jasper Van der Auwera, Erwin Aertbeliën, and Wilm Decré are with the Department of Mechanical Engineering, KU Leuven, 3000 Leuven, Belgium, and also with Flanders Make, 3000 Leuven, Belgium (e-mail: jasper.vanderauwera@kuleuven.be; erwin.aertbelien@kuleuven.be; wilm.decre@kuleuven.be).

Herman Bruyninckx is with the Department of Mechanical Engineering, KU Leuven, 3000 Leuven, Belgium, also with the Department of Mechanical Engineering, TU Eindhoven, 5600 MB Eindhoven, The Netherlands, and also with Flanders Make, 3000 Leuven, Belgium (e-mail: herman.bruyninckx@kuleuven.be).

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918



(a) (b) (c)

Fig. 1. Manual assembly of an Epicyclic Gear Train (EGT), illustrating the type of application considered in this work. (a) High-level plan (assembly plan). Adapted from [6]. (b) At the start of the assembly, the components are distributed over the workspace, such that each (set of) components corresponds to a target T_k with $k = 1 \dots 8$. To decide whether the human operator will deviate from the high-level plan, this work predicts where the human operator is reaching towards (orange, dashed line) based on a partial trajectory (blue, solid line), such that an early intervention in the human's task execution can be provided. Adapted from [6]. (c) The reaching motions executed by the different human operators performing the assembly, which are further explained in section IV.

motion model database.

In section III-C, three approaches are developed which, starting from the motion model database, predict the intended motion target of a human operator for a new, ongoing trajectory. These approaches must also be able to detect when the partial motion is outside the scope of the task. The first approach, classification intent estimation, assesses the similarity between the partial trajectory and each of the motion models in the database, performing probabilistic classification over the different motion classes. This approach requires each model to correspond to a single, predefined target, such that the classification over the motion classes can be mapped to a classification over the predefined targets. The second approach, prediction intent estimation, predicts the end-point of the partial trajectory and evaluates its distance to the predefined targets. As described by Callens et al. [2], this type of motion prediction can be performed by selecting and fitting the most probable TraPPCA model to the partial trajectory. The third approach, mixture intent estimation, also predicts the end-point of the partial trajectory, but uses a mixture of TraPPCA models instead of a single TraPPCA model. Therefore, the TraPPCA models in the database are combined to perform the prediction, which provides additional flexibility.

Finally, in section III-E, this work investigates intent-based decision making for operator assistance in manipulation tasks, where the aim is to decide whether an intervention in the human's task execution is required, based on the predicted target probabilities and the task's high-level plan. Often, the intended target is first selected as the target with the highest probability, after which a decision is made by comparing this target with the high-level plan. However, this approach only uses part of the information from the probabilistic intent estimation. Therefore, this work proposes a decision-making approach which directly uses the estimated target probabilities in the decision-making process, making it more adaptable to different situations and better at dealing with large uncertainties.

In summary, the main contributions of this work are the following: (i) We introduce novel semi-supervised and unsupervised online learning algorithms using TraPPCA and compare them with existing supervised learning for the application of human intent estimation. (ii) We introduce and compare 3 novel reaching motion intent estimation approaches using TraPPCA: (1) classification intent estimation, (2) prediction intent estimation, and (3) mixture intent estimation. (iii) We introduce a fully probabilistic intent-based decision-making approach for operator assistance and compare it with decision making using only the most probable target. The above contributions are validated on the publicly available HA4M dataset [6], containing data collected using a single Azure Kinect camera and markerless skeleton tracking. This markerless tracking provides additional challenges related to accuracy, noise and sample rate, but is closer to a real-world application.

II. Related Work

In literature, different methods have already been proposed for human reaching motion prediction and intent estimation, mostly focusing on HRC. A key part of these methods are the mathematical representations used to describe the human motions, which allow the methods to reason about the future based on a partial, ongoing trajectory. In a first body of research, model-based approaches were used to describe human reaching motions. A well-known method in this area is the Minimum Jerk model [8], which is based on the observation that ideal motions of the human hand in free space are roughly straight with a single-peaked, bell-shaped speed profile. Landi et al. [9] combined this Minimum Jerk model with a Semi-Adaptable Neural Network (S-ANN), aiming to improve the long-term predictions from this model-based approach by incorporating accurate short-term predictions from the S-ANN. Zanchettin and Rocco [10] proposed a different model-based approach, which predicts the future position of the human's hand by minimizing the overall curvature of the motion. This model-based trajectory generator was combined with a Bayesian inference engine to predict the human hand reaching target. However, these model-based representations are generally too restrictive to describe the variety of reaching motions encountered in real manipulation tasks.

Alternatively, Inverse Optimal Control (IOC) has also been used in literature to learn a cost function for human reaching motions. Mainprice et al. [11] employed IOC to determine the cost function for single-arm reaching motions and applied this cost function in the STOMP algorithm for human motion prediction through iterative replanning. However,

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

this approach required high-dimensional data (23 DoFs) and needed velocity, acceleration and jerk to be calculated through finite differencing, which is not feasible for more noisy data. Oguz et al. [12] also used IOC to gain understanding about human reaching motions and combined it with Probabilistic Movement Primitives (ProMP) [13] to perform predictions. However, they made the assumption that shoulder translations and torso movements can be ignored, which is not necessarily the case for real reaching motions.

Imitation learning has been employed as well to obtain probabilistic motion models for human reaching motion prediction and intent estimation. Mainprice and Berenson [14] used Gaussian Mixture Models (GMMs) [15] to estimate the workspace occupancy of a human in close proximity to a robot. To achieve this, the swept volume of the human reaching motions was predicted online using an offline learned library of GMMs. Similarly, Pérez-D'Arpino and Shah [16] employed a data-driven approach to construct a library of motion models offline, which was then used for online prediction of the intended human reaching target. Furthermore, this work investigated using additional task-level information as a prior for human reaching motion prediction. However, the demonstrations used to build model libraries were recorded from the same person who later executed the task, not yet building a common library for multiple people. Luo et al. [17] [18] introduced an unsupervised online learning algorithm for GMMs together with a two-layer framework for human reaching motion prediction. The unsupervised online learning approach aimed to build motion models on-the-fly, requiring no offline training phase or manually-labeled training data, allowing their framework to adapt to different people and varying motion styles.

Human motion prediction and intent estimation have also been investigated for applications beyond human reaching motions. To provide assistance during manual robot guidance and relieve the user from excessive physical efforts, Nicolis et al. [1] developed a framework of two Recurrent Neural Networks to predict the user motion and infer the most probable target, combining the neural network output with a standard admittance controller. To provide assistance with an exoskeleton, Callens [2] proposed a framework for motion onset detection, progress speed estimation, intent recognition and motion prediction based on an offline learned motion model database, applying the framework to squatting, stooping and lifting actions and comparing the performance using TraPPCA and ProMP. To provide assistance during teleoperated robot manipulation, Song [19] introduced a shared-control paradigm that leverages a deep learning-based prediction approach called Robot Trajectron to anticipate the motion of the robot's end-effector.

This work focuses on industrial contexts, where rapid deployment of new tasks is important and high-level plans of tasks are often documented and readily available, such that learning-based methods [20] are not required to determine these relations. However, model-based information about the reaching motions and locations required to execute these plans is typically not available, requiring this information to be learned from demonstrations. These considerations have lead to the following, more unique aspects of this work.

The motion models in this work use an explicit progress dependency to facilitate reasoning about the future, which is beneficial for predicting the intended motion target and its influence on the high-level plan. This focuses the work on imitation learning approaches like ProMP [13] and TraPPCA [7]. However, the explicit progress dependency also introduces the additional challenge of estimating progress along partially executed reaching motions. Here, this challenge is addressed by adapting the Dynamic Time Warping (DTW) algorithm [21], a dynamic programming algorithm to find the optimal alignment between two time series.

To enable quick deployment, the number of training demonstrations must be small. The ability of ProMP and TraPPCA to handle limited demonstrations was already shown in literature for other applications. Callens [22] learned human motion models using as few as 10 demonstrations, considering actions like squatting, stooping and lifting. Other works learned motion models for robot actions using 5 [23] and 10 [13] human demonstrations. Furthermore, a similar number of training demonstrations was used by Luo et al. [17] [18] to model human reaching motions with GMMs, which demonstrates that a decent baseline representation for human reaching motions can be obtained using limited training data.

Previous work has employed imitation learning approaches like ProMP and TraPPCA for offline supervised learning [2]. However, since human task performances may change over time (e.g., due to fatigue) and may vary across different operators, this work also investigates semi-supervised [24] and unsupervised online learning approaches and their ability to deal with these variations. Finally, this work investigates how probabilistic intent estimation can be combined more effectively with a predefined high-level plan for intent-based decision making, improving operator assistance in manipulation tasks.

III. Methods

Subsection III-A introduces the modeling approach that is used to describe a single motion type. All motion types are stored in a motion model database $\mathcal{L}_M$, consisting of K motion models M_j. Then, subsection III-B describes the similarity measures used to evaluate whether a complete or partial trajectory corresponds to a certain motion model, enabling probabilistic classification within $\mathcal{L}_M$. Next, subsection III-C describes three different approaches to identify the intent (i.e. the goal/target) of a partial motion, after which subsection III-D describes how the motion model database $\mathcal{L}_M$ used during intent estimation can be learned or updated online, during execution. Finally, subsection III-E describes how these different approaches are combined in an intent-based decision-making approach.

In general, the relationship between motion models and targets is flexible. Multiple motion models may describe different approaches towards the same target, such as left and right paths around an obstacle. A single motion model may also describe motions towards multiple targets. Without loss of generality, the remainder of this work assumes that the intent of a partial motion is described by the motion's target location.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

A. Learning and Prediction for a Single Motion Model

1) Pre-Processing: Before a motion model $M_j \in \mathcal{L}_M$ is learned from a set of training demonstrations, each demonstration in the set undergoes pre-processing. The first step is to reparameterize each demonstration $\boldsymbol{f}_{\text{dem}}$ in function of a progress variable $s_i \in [0, 1]$, with $s_1 = 0$ corresponding to the first sample and $s_D = 1$ to the final sample of the recorded demonstration. This reparameterization is done to improve alignment between the training demonstrations. A common choice is normalized time:

$$s_i = \frac{t_i - t_1}{T}, \tag{1}$$

where t_1 is the time of recording of the first sample, t_i is the time of recording of sample i and T is the duration of the demonstration. This reparameterization maintains the (scaled) velocity profile along the trajectory, but if this velocity profile is not important for the application, a reparameterization based on the scaled path length can also be considered. Afterwards, each demonstration is resampled to points $f_{\text{dem},n}$ with $n = 1 \dots N$ using a linear interpolation between the recorded points, such that all training demonstrations have the same number of samples at uniform progress steps.

2) TraPPCA Model: As described by Aertbeliën and De Schutter [7], a d-dimensional trajectory consisting of N progress samples is rearranged in a single column vector $\bar{\boldsymbol{f}} \in \mathbb{R}^{dN}$, after which Probabilistic Principal Component Analysis is applied to obtain the following linear latent variable model for a set of N_d demonstrations:

$$\bar{\boldsymbol{f}} = \bar{\boldsymbol{H}}\boldsymbol{\chi} + \bar{\boldsymbol{b}} + \bar{\boldsymbol{\epsilon}}. \tag{2}$$

Here, $\bar{\boldsymbol{H}} \in \mathbb{R}^{dN \times m}$ is a matrix containing m basis functions, which encode the variability in the demonstrations and are weighted by the latent variable vector $\boldsymbol{\chi} \in \mathbb{R}^m$, distributed as:

$$\boldsymbol{\chi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m), \tag{3}$$

with $\boldsymbol{I}_m$ an $m \times m$ identity matrix. Furthermore, $\bar{\boldsymbol{b}} \in \mathbb{R}^{dN}$ represents the mean trajectory of the motion model and $\bar{\boldsymbol{\epsilon}} \in \mathbb{R}^{dN}$ is a Gaussian noise vector:

$$\bar{\boldsymbol{\epsilon}} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_{dN}) \tag{4}$$

with variance σ^2. For the application in this paper, the number of dimensions $d = 3$ and the degrees of freedom consist of the x, y and z coordinates of a trajectory. However, the approach is not limited to this choice of degrees of freedom.

For a given number of latent variables m, the values for $\bar{\boldsymbol{H}}$, $\bar{\boldsymbol{b}}$ and σ are calculated from the dataset employing maximum likelihood, using the eigenvalues and eigenvectors of the sample covariance matrix of $\bar{\boldsymbol{f}}$. This maximum likelihood approach and the choice of m are further explained in [7].

3) Prediction With Single TraPPCA Model: The above model, without further prior knowledge, implies that $\bar{\boldsymbol{f}} \sim \mathcal{N}(\bar{\boldsymbol{b}}, \bar{\boldsymbol{H}}\bar{\boldsymbol{H}}^\top + \sigma^2 \boldsymbol{I}_{dN})$, since $\boldsymbol{\chi}$ and $\bar{\boldsymbol{\epsilon}}$ are independent. This model can be conditioned on a series of observations $\boldsymbol{f}_{\text{obs},i}$ with $i = 1 \dots O$, where a Kalman filter update step [25] is iteratively applied for each observation to update the estimate of the latent variable vector $\boldsymbol{\chi}_{\text{cond}} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{cond}}, \boldsymbol{\Sigma}_{\text{cond}})$, with $\boldsymbol{\mu}_{\text{cond}}$ the state and $\boldsymbol{\Sigma}_{\text{cond}}$ the state covariance of the Kalman filter. Since the progress values s_i of the observations $\boldsymbol{f}_{\text{obs},i}$ do not correspond to the progress samples used within the TraPPCA model, the matrix functions $\boldsymbol{H}(s) \in \mathbb{R}^{d \times m}$ and $\boldsymbol{b}(s) \in \mathbb{R}^d$ are introduced by fitting a cubic spline through the appropriate elements of $\bar{\boldsymbol{H}}$ and $\bar{\boldsymbol{b}}$, respectively. The linear latent variable model of (2) is then rewritten as:

$$\boldsymbol{f}(s) = \boldsymbol{H}(s)\boldsymbol{\chi} + \boldsymbol{b}(s) + \boldsymbol{\epsilon}, \tag{5}$$

with $\boldsymbol{f}(s) \in \mathbb{R}^d$ and $\boldsymbol{\epsilon} \in \mathbb{R}^d$.

To attach more importance to recent observations, process noise $\boldsymbol{\Sigma}_{\text{process}} = \sigma^2_{\text{process}} \boldsymbol{I}_m$ is added to the prediction step of the Kalman filter approach. The ratio between $\boldsymbol{\Sigma}_{\text{process}}$ and the uncertainty $\boldsymbol{\Sigma}_{f_{\text{obs},i}}$ on observations $\boldsymbol{f}_{\text{obs},i}$ determines the dynamics of the forgetting. The employed Kalman filter approach is elaborated in Algorithm 1 of Supplemental Material.

4) End-Point Prediction: Given the latent variable vector $\boldsymbol{\chi}_{\text{cond}} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{cond}}, \boldsymbol{\Sigma}_{\text{cond}})$, a prediction of the target position can be made using the TraPPCA model (2). This results in a Gaussian end-point prediction with following mean and covariance:

$$\boldsymbol{\mu}_{\text{end}} = \boldsymbol{H}(1)\boldsymbol{\mu}_{\text{cond}} + \boldsymbol{b}(1) \tag{6}$$

$$\boldsymbol{\Sigma}_{\text{end}} = \boldsymbol{H}(1)\boldsymbol{\Sigma}_{\text{cond}}\boldsymbol{H}(1)^\top + \sigma^2 \boldsymbol{I}_d \tag{7}$$

Here, $\boldsymbol{H}(1)$ and $\boldsymbol{b}(1)$ are the matrix functions of (5) evaluated at $s = 1$ and σ^2 characterizes the Gaussian noise vector of the TraPPCA model.

B. Probabilistic Trajectory Classification Over Model Library

1) Model-Trajectory Similarity Assessment: To assess the similarity between a TraPPCA model $M_j \in \mathcal{L}_M$ and a collection of observations parameterized in time $\boldsymbol{obs}_t = \left\{(\boldsymbol{f}_{\text{obs},i}, t_i)\right\}_{i=1}^{O}$ corresponding to a full trajectory, the observations are first pre-processed as described in section III-A1, resulting in observations parameterized in progress $\boldsymbol{obs}_s = \left\{(\boldsymbol{f}_{\text{obs},n}, s_n)\right\}_{n=1}^{N}$. Then, the distance between each progress sample coordinate $\boldsymbol{f}_{\text{obs},n} \in \mathbb{R}^d$ and the corresponding sample $\bar{\boldsymbol{f}}_{j,n}$ of model M_j is computed as the squared Mahalanobis distance (SMD):

$$\text{SMD}[\boldsymbol{f}_{\text{obs},n}, \bar{\boldsymbol{f}}_{j,n}] = (\boldsymbol{f}_{\text{obs},n} - \bar{\boldsymbol{b}}_{j,n})^\top \bar{\boldsymbol{\Sigma}}^{-1}_{j,n} (\boldsymbol{f}_{\text{obs},n} - \bar{\boldsymbol{b}}_{j,n}), \tag{8}$$

with

$$\bar{\boldsymbol{f}}_{j,n} \sim \mathcal{N}(\bar{\boldsymbol{b}}_{j,n}, \bar{\boldsymbol{\Sigma}}_{j,n}) = \mathcal{N}(\bar{\boldsymbol{b}}_{j,n}, \bar{\boldsymbol{H}}_{j,n}\bar{\boldsymbol{H}}^\top_{j,n} + \sigma^2 \boldsymbol{I}_d). \tag{9}$$

Based on this, the total distance $D_j(\boldsymbol{obs}_s)$ between the model M_j and the full collection of progress samples $\boldsymbol{obs}_s$ is calculated as the sum of the distances $\text{SMD}[\boldsymbol{f}_{\text{obs},n}, \bar{\boldsymbol{f}}_{j,n}]$ corresponding to each sample $\boldsymbol{f}_{\text{obs},n}$:

$$D_j(\boldsymbol{obs}_s) = \sum_{n=1}^{N} \text{SMD}[\boldsymbol{f}_{\text{obs},n}, \bar{\boldsymbol{f}}_{j,n}] \tag{10}$$

Note that this summed squared Mahalanobis distance is only a heuristic approximation. It provides a distance measure which takes into account the correlations between the dimensions of each sample, as encoded in the covariance matrices $\bar{\boldsymbol{\Sigma}}_{j,n}$, but neglects the covariance between the different observations $\boldsymbol{f}_{\text{obs},n}$. This approximation will allow us to efficiently apply Dynamic Time Warping in the following section.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

2) Dynamic Time Warping for Partial Trajectories: Unlike complete trajectories, the progress values of a partial trajectory cannot be calculated as described in section III-A1, since the duration of the motion is not yet known. Therefore, to assess the similarity between a partial trajectory and a model M_j, similarly as described in section III-B1, the progress of the partial trajectory must first be estimated. Here, this is done using Dynamic Time Warping (DTW) [21].

DTW is an algorithm commonly used to find the optimal alignment between two time series with different lengths and varying time axes, aiming to minimize the deviation between them based on a selected distance metric. In this work, a probabilistically informed adaptation of this algorithm is used to find the optimal alignment between a partial trajectory $\boldsymbol{obs}_t = \left\{(\boldsymbol{f}_{\text{obs},i}, t_i)\right\}_{i=1}^{O}$ and a TraPPCA model M_j. In this algorithm, the squared Mahalanobis distance $\text{SMD}[\boldsymbol{f}_{\text{obs},i}, \bar{\boldsymbol{f}}_{j,n}]$ between observed sample $\boldsymbol{f}_{\text{obs},i}$ and model sample $\bar{\boldsymbol{f}}_{j,n}$ is used as the distance metric. The outputs of this DTW algorithm are the summed squared Mahalanobis distance $D_j(\boldsymbol{obs}_t)$, the current progress $s_{\text{curr},j}$, and the warping path $\boldsymbol{w}_j = (\hat{\boldsymbol{i}}_j, \hat{\boldsymbol{n}}_j)$ with path length L_j corresponding to the optimal alignment between the partial trajectory and the model. In the warping path, $\hat{\boldsymbol{i}}_j$ contains the sample indices i of the partial trajectory along the warping path and $\hat{\boldsymbol{n}}_j$ contains the sample indices n of the model along the warping path. A detailed description of this algorithm is included in Supplemental Material.

3) Probabilistic Classification of Partial Trajectories: Once the summed squared Mahalanobis distances $D_j(\boldsymbol{obs}_t)$ are calculated for each motion model M_j, they are first used to determine whether the observed partial trajectory corresponds to any of the motion types encoded in the motion model database. This is achieved by performing an hypothesis test [26] for each motion model M_j, assessing the null hypothesis of $\boldsymbol{obs}_t$ being explained by M_j. Since the summed squared Mahalanobis distance follows a χ^2-distribution with dL_j degrees of freedom, the critical value D_j^c for rejecting the null hypothesis is obtained from this distribution using a selected significance level α for the right-tail probability (Fig. 2 in Supplemental Material). When the observed distance $D_j(\boldsymbol{obs}_t)$ exceeds the critical value D_j^c, it indicates that the observed trajectory does not correspond to the motion model M_j, given risk α of wrongfully making this decision. If this rejection of the null hypothesis takes place for all motion models in the database, $\boldsymbol{obs}_t$ is classified as an "other" motion, which is not (yet) included in the motion model database. When many motion models are considered, the effect of multiple comparisons must be taken into account by choosing the significance level α with a Bonferroni correction [27].

If the null hypothesis is not rejected for all motion models in the database, the probability distribution $P(M_j|\boldsymbol{obs}_t)$ over the motion models is determined. Since the length of the warping path L_j can vary for the different models M_j (cf. section III-B2) and the likelihood of the observed coordinates $\boldsymbol{obs}_t$ should not depend on the length of the warping path, this work compensates for this length by taking the geometric mean of the probability densities at each element of the warping path:

$$P(\boldsymbol{obs}_t|M_j) = \left[\prod\nolimits_{(i,n)\in\boldsymbol{w}_j} p(\boldsymbol{f}_{\text{obs},i}|\bar{\boldsymbol{f}}_{j,n})\right]^{1/L_j}, \tag{11}$$

with

$$p(\boldsymbol{f}_{\text{obs},i}|\bar{\boldsymbol{f}}_{j,n}) = \frac{exp(-0.5\cdot \text{SMD}[\boldsymbol{f}_{\text{obs},i}, \bar{\boldsymbol{f}}_{j,n}])}{\sqrt{(2\pi)^d\left|\bar{\boldsymbol{\Sigma}}_{j,n}\right|}}. \tag{12}$$

This leads to the following expression for the likelihood $P(\boldsymbol{obs}_t|M_j)$ of $\boldsymbol{obs}_t$ given model M_j:

$$\begin{aligned} P(\boldsymbol{obs}_t|M_j) &= \left[C_j\cdot exp(-0.5\cdot D_j(\boldsymbol{obs}_t))\right]^{1/L_j} \\ &= C_j^{1/L_j}\cdot exp(-0.5\cdot D_j(\boldsymbol{obs}_t)/L_j), \end{aligned} \tag{13}$$

where C_j depends on the DTW alignment, thus also on $\boldsymbol{obs}_t$:

$$C_j = \left[\prod\nolimits_{i=1}^{L_j}\sqrt{(2\pi)^d|\bar{\boldsymbol{\Sigma}}_{j,\hat{n}_{j,i}}|}\right]^{-1}, \tag{14}$$

with $\bar{\boldsymbol{\Sigma}}_{j,\hat{n}_{j,i}}$ representing the covariance matrix of the model sample corresponding to the i-th element of the warping path.

This geometric mean was also used by Luo et al. [17] [18] to compensate for the number of elements when calculating the probability $P(\boldsymbol{obs}_t|G_j)$ of a trajectory $\boldsymbol{obs}_t$ for each GMM G_j in a motion model library. The aim there was to compare these likelihoods with a constant threshold regardless of the trajectory length, such that a compensation for the trajectory length was required.

Once the likelihoods are calculated, they are used to determine the posterior probabilities $P(M_j|\boldsymbol{obs}_t)$ of each motion model M_j using Bayes' rule:

$$P(M_j|\boldsymbol{obs}_t) = \frac{P(\boldsymbol{obs}_t|M_j)P(M_j)}{\sum_k P(\boldsymbol{obs}_t|M_k)P(M_k)} \tag{15}$$

Here, $P(M_j)$ corresponds to the prior probability of model M_j, which is selected based on prior knowledge about the application. If no prior knowledge is available, equal priors are selected, causing them to disappear from equation (15).

4) Probabilistic Classification of Complete Trajectories: For complete trajectories, a similar procedure as described in section III-B3 is performed. However, the summed squared Mahalanobis distances $D_j(\boldsymbol{obs}_s)$ of complete trajectories are directly determined as described in section III-B1, starting from the pre-processed trajectory $\boldsymbol{obs}_s$, which has the following influence on the calculations performed during the probabilistic classification:

- For the hypothesis test, $D_j(\boldsymbol{obs}_s)$ now follows a χ^2-distribution with dN degrees of freedom. Since N is constant for a given trajectory $\boldsymbol{obs}_s$, the same distribution and corresponding critical value D_j^c are found for each motion model M_j.
- For the calculation of the probability distribution $P(M_j|\boldsymbol{obs}_s)$ over the motion models M_j, the likelihoods are now determined using the following function:

$$P(\boldsymbol{obs}_s|M_j) = \left[C_j\cdot exp(-0.5\cdot D_j(\boldsymbol{obs}_s))\right]^{1/N} \tag{16}$$

where C_j is a constant given by:

$$C_j = \left[\prod\nolimits_{n=1}^{N}\sqrt{(2\pi)^d\left|\bar{\boldsymbol{\Sigma}}_{j,n}\right|}\right]^{-1} \tag{17}$$

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

and $\bar{\boldsymbol{\Sigma}}_{j,n}$ represents the covariance matrix of the n-th model sample $\bar{\boldsymbol{f}}_{j,n}$.

C. Intent Estimation

In this work, intent estimation refers to the prediction of the intended reaching target based on a partial execution of the corresponding trajectory. The intent estimation approaches presented in this work first decide whether the observed partial trajectory $\boldsymbol{obs}_t$ approaches any of the predefined targets in the workspace, aiming to filter out motions that are not part of the current task. If the decision is made that a known target is being approached, the probability distribution $P(T_k|\boldsymbol{obs}_t)$ over the predefined targets T_k is calculated, indicating the human's intent. However, if the decision is made that no known target is being approached, the motion is classified as "other" and no probability distribution is calculated.

Below, three intent estimation approaches are introduced, which are based on the building blocks described in the previous sections.

1) Classification Intent Estimation: In this first intent estimation approach, the intended target of a partial reaching motion $\boldsymbol{obs}_t$ is predicted by evaluating the similarity between this partial motion and the available motion models $M_j \in \mathcal{L}_M$. Before the similarity between $\boldsymbol{obs}_t$ and M_j is calculated, this approach starts by increasing the covariance of each motion model M_j with a term $\sigma^2_{\text{ADD,M}}\boldsymbol{I}_{dN}$, where $\sigma_{\text{ADD,M}}$ is a tuning parameter. The aim of this step is to compensate for the additional variability that may occur in the modeled motion class, but was not yet observed in the training demonstrations. Then, this approach uses the DTW procedure described in sections III-B2 and III-B3, which either classifies the partial trajectory as "other" or provides the posterior probability $P(M_j|\boldsymbol{obs}_t)$ for each motion model M_j, depending on the similarity between $\boldsymbol{obs}_t$ and the motion models encoded in $\mathcal{L}_M$. However, to be able to transfer these posterior probabilities to the desired probabilities $P(T_k|\boldsymbol{obs}_t)$ of the predefined targets T_k, the employed motion model database $\mathcal{L}_M$ must be constructed such that each motion model M_j corresponds to a single reaching target T_k.

2) Prediction Intent Estimation: In this approach, intent estimation is performed by determining the motion model that best corresponds to the observed partial trajectory, predicting the end-point of the partial trajectory using the selected model and evaluating the distance between the predicted end-point and the predefined targets. To identify the motion model $M_j \in \mathcal{L}_M$ that best corresponds to the observed partial trajectory $\boldsymbol{obs}_t$, this approach starts by performing the same steps as previously described for classification intent estimation. If the partial trajectory is classified as "other", no end-point prediction is performed and the procedure ends. Otherwise, the best corresponding motion model is selected as the model with the highest posterior probability $P(M_j|\boldsymbol{obs}_t)$.

Then, the selected motion model is conditioned on the partial trajectory, as described in section III-A3, which requires the progress values $\boldsymbol{s} \in \mathbb{R}^O$ for the partial trajectory to be known. These progress values are estimated based on the optimal alignment between the partial trajectory and the selected motion model, which is already available from the DTW procedure performed in the previous model selection steps. This is achieved by fitting the progress model of section III-A1 to the time-progress relation obtained from this optimal alignment. Afterwards, a Gaussian end-point prediction $\mathcal{N}(\boldsymbol{\mu}_{\text{end}}, \boldsymbol{\Sigma}_{\text{end}})$ is determined as described in section III-A4.

Finally, the proximity between this Gaussian end-point prediction and each predefined target T_k is evaluated using the squared Mahalanobis distance (8). However, before these distances are calculated, the covariance matrix of the end-point prediction is increased with a term $\sigma^2_{\text{ADD,P}}\boldsymbol{I}_d$. The aim of this step is to account for additional sources of variability present during the proximity assessment (e.g. uncertainty in the target locations). Then, a similar procedure as described in section III-B4 is performed to either classify the predicted end-point as "other" or provide the posterior probability $P(T_k|\boldsymbol{obs}_t)$ for each target T_k, depending on the proximity between the predicted end-point and the predefined targets. Since this comparison is only based on a single squared Mahalanobis distance per target, the number of distances N is equal to 1 in this procedure.

3) Mixture Intent Estimation: In this approach, intent estimation is also performed by predicting the end-point of the observed partial trajectory and evaluating its distance with respect to each predefined target. However, instead of selecting a single motion model to perform the prediction, this approach combines the TraPPCA models in the motion model database to obtain a single end-point prediction.

The approach starts by determining the optimal alignment and corresponding summed squared Mahalanobis distance between the partial trajectory $\boldsymbol{obs}_t$ and each motion model $M_j \in \mathcal{L}_M$ using the DTW algorithm of section III-B2, applying the same steps as previously described for classification intent estimation. Then, the hypothesis test of section III-B3 is performed for each motion model M_j, to determine whether the observed partial trajectory should be classified as "other". If this is the case, no end-point prediction is performed and the procedure ends.

If the partial trajectory is not classified as "other", the approach continues by estimating the progress values $\boldsymbol{s}_j \in \mathbb{R}^O$ of the partial trajectory $\boldsymbol{obs}_t$ for each motion model M_j. This is again achieved by fitting the progress model of section III-A1 to the time-progress relation obtained from the DTW algorithm. Afterwards, each TraPPCA model in the database is conditioned on the observed partial trajectory, as described in section III-A3, where the estimated progress values $\boldsymbol{s}_j$ are used as the progress of the partial trajectory. This results in a latent variable vector $\boldsymbol{\chi}_{\text{cond},j} \sim \mathcal{N}(\boldsymbol{\mu}_{\text{cond},j}, \boldsymbol{\Sigma}_{\text{cond},j})$ for each motion model in the database, which allows to determine a probabilistic end-point prediction $\mathcal{N}(\boldsymbol{\mu}_{\text{end},j}, \boldsymbol{\Sigma}_{\text{end},j})$ for each model, as described in section III-A4.

Next, the weights of the different TraPPCA models are determined. These weights correspond to the responsibility that each model M_j takes for explaining the observation $\boldsymbol{obs}_t$ [28], which is determined as:

$$h_j(\boldsymbol{obs}_t) = \frac{P(M_j)\mathcal{N}(\boldsymbol{obs}_t|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} P(M_k)\mathcal{N}(\boldsymbol{obs}_t|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \tag{18}$$

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

Here, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ represent the mean and covariance of model M_j after conditioning on $\boldsymbol{obs}_t$:

$$\boldsymbol{\mu}_j = \boldsymbol{H}_j(\boldsymbol{s}_j)\boldsymbol{\mu}_{\text{cond},j} + \boldsymbol{b}_j(\boldsymbol{s}_j) \tag{19}$$

$$\boldsymbol{\Sigma}_j = \boldsymbol{H}_j(\boldsymbol{s}_j)\boldsymbol{\Sigma}_{\text{cond},j}\boldsymbol{H}_j(\boldsymbol{s}_j)^\top + (\sigma_j^2 + \sigma_{\text{ADD,M}}^2)\boldsymbol{I}_{dO} \tag{20}$$

with σ_j determining the Gaussian noise component of model M_j, $\sigma_{\text{ADD,M}}$ introducing additional variability with respect to M_j and $\boldsymbol{H}_j(\boldsymbol{s}_j)$ and $\boldsymbol{b}_j(\boldsymbol{s}_j)$ representing the matrix functions obtained by fitting a cubic spline through the appropriate elements of $\bar{\boldsymbol{H}}_j$ and $\bar{\boldsymbol{b}}_j$, evaluated at the estimated progress values $\boldsymbol{s}_j$. Furthermore, $P(M_j)$ indicates the prior probability of model M_j, which is selected based on prior knowledge about the application. If no prior knowledge is available, equal priors are selected, causing them to disappear from equation (18). The numerical stability of these weight calculations can be improved similarly to the softmax function [29], since the Gaussian distribution is also exponential.

Finally, the end-point $\boldsymbol{f}_{\text{end}}$ of $\boldsymbol{obs}_t$ is predicted using following mixture distribution:

$$P(\boldsymbol{f}_{\text{end}}|\boldsymbol{obs}_t) \sim \sum_{j=1}^{K} h_j(\boldsymbol{obs}_t)\mathcal{N}(\boldsymbol{\mu}_{\text{end},j}, \boldsymbol{\Sigma}_{\text{end},j}) \tag{21}$$

However, to be able to calculate the squared Mahalanobis distance between the probabilistic end-point prediction and each predefined target, the end-point prediction must be Gaussian. Therefore, according to Calinon's Gaussian Mixture Regression (GMR) [30], the mean and covariance of the mixture are calculated, obtaining:

$$\boldsymbol{\mu}_{\text{end}}^{\text{mix}} = \sum_{j=1}^{K} h_j(\boldsymbol{obs}_t)\boldsymbol{\mu}_{\text{end},j} \tag{22}$$

$$\boldsymbol{\Sigma}_{\text{end}}^{\text{mix}} = \sum_{j=1}^{K} h_j(\boldsymbol{obs}_t)\left(\boldsymbol{\Sigma}_{\text{end},j} + \boldsymbol{\mu}_{\text{end},j}\boldsymbol{\mu}_{\text{end},j}^\top\right) - \boldsymbol{\mu}_{\text{end}}^{\text{mix}}\boldsymbol{\mu}_{\text{end}}^{\text{mix}\top} \tag{23}$$

Once this Gaussian end-point prediction is obtained, the same procedure as described in section III-C2 is performed to assess the proximity between the predicted end-point and each predefined target.

It is important to note that this approach assumes a continuous workspace between the motions encoded in the motion model database, such that all linear combinations of the encoded motions also provide a feasible motion. Using this approach, end-point predictions can be obtained for a broader range of reaching motions, providing additional flexibility during intent estimation. However, it also entails that possible constraints (e.g. obstacles) implicitly encoded in the motion model database are not necessarily taken into account during intent estimation.

D. Online Learning

In order to perform intent estimation, as described in section III-C, the human motion model database $\mathcal{L}_M$ must first be learned from a set of complete human motion trajectories. In previous work [2], this motion model database was constructed offline using supervised learning. This approach started from manually-labelled training demonstrations, where the labels indicate the purpose of the motion, and learned a motion model for each group of demonstrations with the same label. However, a disadvantage of this approach is that the model database cannot be adapted to encode new motion styles observed during execution, since each new demonstration must first be labelled offline.

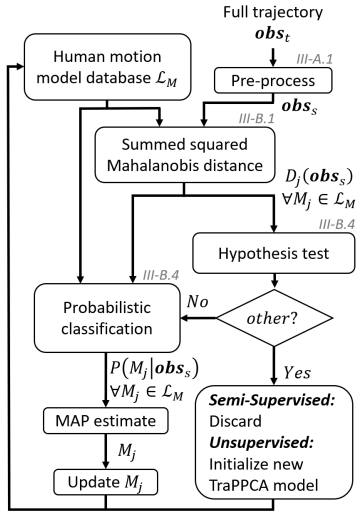


Fig. 2. Workflow for the semi-supervised and unsupervised online learning approaches.

Alternatively, online learning approaches can be used to build or adapt the motion model database during execution. In these approaches, each new demonstration is labelled online in an unsupervised fashion, without human input, such that the demonstration can be directly encoded in the motion model database. To achieve this, two novel online learning approaches using TraPPCA are proposed in this section, based on semi-supervised online (SSO) and unsupervised online (UO) learning. The general workflow for both approaches is visualized in Fig. 2, using the building blocks discussed in the previous sections.

1) Semi-Supervised Online Learning: The proposed semi-supervised online learning approach initializes the motion model database through offline supervised learning, resulting in a predefined number of motion models, which are then updated online with each new, complete demonstration. To achieve this, each new, time-dependent trajectory $\boldsymbol{obs}_t$ observed during execution is first pre-processed as described in section III-A1, resulting in the progress-dependent trajectory $\boldsymbol{obs}_s$. Next, the summed squared Mahalanobis distance $D_j(\boldsymbol{obs}_s)$ is calculated for each motion model $M_j \in \mathcal{L}_M$, as described in section III-B1. However, to compensate for the additional variability that may occur in the modelled motion class, but was not yet observed in the training demonstrations, the covariance of each motion model is increased with a term $\sigma_{\text{ADD,M}}^2\boldsymbol{I}_{dN}$ before calculating this summed squared Mahalanobis distance (see also section III-C1). These distances are then used to perform the hypothesis test of section III-B4, determining whether the trajectory should be classified as "other". If classified as "other", $\boldsymbol{obs}_s$ does not correspond to any of the motion types encoded in the motion model database and the trajectory is discarded. Otherwise, the probabilistic

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

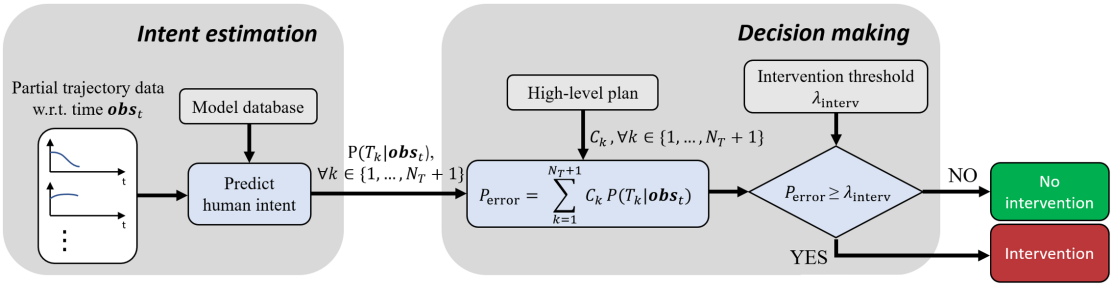


Fig. 3. Overview of the intent-based decision-making approach using error probability P_{error} at observed sample O. In the intent estimation block, the target probabilities $P(T_k|\boldsymbol{obs}_t)$ are determined using the intent estimation approaches of section III-C, which employ a model database obtained from the learning approaches of section III-D.

classification of section III-B4 is performed, providing the posterior probability $P(M_j|\boldsymbol{obs}_s)$ for each model $M_j \in \mathcal{L}_M$. Finally, the label of the trajectory is determined as the maximum a posteriori (MAP) estimate over the calculated posterior probabilities, the trajectory is added to the training data of the corresponding motion class and the model is re-learned.

2) Unsupervised Online Learning: Unlike the semi-supervised online learning approach, the proposed unsupervised online learning approach starts from an empty motion model database, not requiring offline training or manually-labelled data, and uses each new, complete demonstration to either update an existing model or initialize a new model, depending on its similarity to the existing models. To achieve this, a similar workflow is employed as described for the semi-supervised online learning approach. However, when the motion model database is empty (first trajectory) or the trajectory is classified as "other", the trajectory is now used to initialize a new TraPPCA model. Since TraPPCA cannot learn a probabilistic model from a single demonstration, due to the lack of information about the variability of the learned motion, a first estimate of the expected variability must be introduced during the initialization process. To do so, random variations with respect to the available demonstration are generated as described by Luo et al. [17], [18], constructing a set of training demonstrations located within a predefined distance from the initial demonstration. The advantage of this approach is that the standard maximum likelihood procedure can still be used to learn the TraPPCA model.

When the trajectory is not classified as "other", the best corresponding motion model is determined using the same procedure as described for the semi-supervised online learning approach. The trajectory is added to the training data of the corresponding motion class, one of its random trajectories generated during initialization is removed (until none are left) and its TraPPCA model is re-learned.

An important remark for both online learning approaches is that the desired number of latent variables for learning a TraPPCA model must be provided beforehand. However, the desired number might not always be feasible, depending on the number and nature of the demonstrations. Therefore, to make sure that each motion model can be learned, this work uses a recursive procedure to learn TraPPCA models, which starts from a desired number of latent variables $m_{\max}$ and decreases this number until the learning procedure is successful.

E. Intent-Based Decision Making

This section focuses on intent-based decision making for operator assistance in manipulation tasks. The aim is to decide whether an intervention in the human's task execution is required, based on the target probabilities $P(T_k|\boldsymbol{obs}_t)$ obtained from intent estimation and the high-level plan of the task. A common approach is to divide the decision-making process into two separate decisions: (1) select the target with the highest probability as the intended target and (2) compare this intended target with the task's high-level plan to decide whether an intervention is required. However, by first selecting the most probable target, only part of the information provided by the probabilistic intent estimation is used during the final decision-making process, since the probability distribution over the targets is not directly considered.

This work proposes an alternative decision-making approach (Fig. 3), which directly integrates the estimated target probabilities in the decision-making process. First, the probability of the human operator deviating from the high-level plan is calculated:

$$P_{\text{error}} = \sum_{k=1}^{N_T+1} C_k P(T_k|\boldsymbol{obs}_t), \tag{24}$$

with $C_k \in \{0, 1\}$ determined by the high-level plan as shown in Fig. 4a, N_T the number of predefined targets and $P(T_k|\boldsymbol{obs}_t)$ the posterior probability of target T_k given partial trajectory $\boldsymbol{obs}_t$. To indicate whether the partial trajectory is classified as "other", an additional probability $P(T_{N_T+1}|\boldsymbol{obs}_t)$ is used, with $P(T_{N_T+1}|\boldsymbol{obs}_t) = 1$ if $\boldsymbol{obs}_t$ is classified as "other" and $P(T_{N_T+1}|\boldsymbol{obs}_t) = 0$ otherwise. Note that, by using equation (24) to calculate this error probability, the assumption is made that each target is either in accordance with or in conflict with the high-level plan, without preferences between the targets in each category. Then, the error probability is compared with an intervention threshold λ_{interv}, which has a value between 0 and 1. If the error probability exceeds the threshold, an intervention is required. If not, the process is repeated for the next update of the partial trajectory.

An important advantage of the proposed approach is that the decision-making process can be adapted to different situations through the selection of the threshold. However, selecting a proper threshold can also be challenging, since it requires further insight into the application. Therefore, this threshold is studied in more detail in section IV-D, together with a comparison of both approaches.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

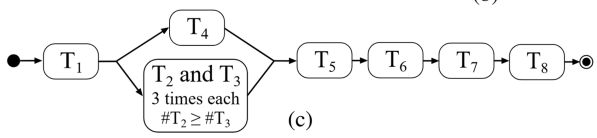
This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

Target	Condition	True	False
T_1	$\#T_1 = 0$	$C_1 = 0$	$C_1 = 1$
T_2	$\#T_1 = 1, \#T_2 < 3$	$C_2 = 0$	$C_2 = 1$
T_3	$\#T_2 > \#T_3$	$C_3 = 0$	$C_3 = 1$
T_4	$\#T_1 = 1, \#T_4 = 0$	$C_4 = 0$	$C_4 = 1$
T_5	$\#T_4 = 1, \#T_3 = 3, \#T_5 = 0$	$C_5 = 0$	$C_5 = 1$
T_6	$\#T_5 = 1, \#T_6 = 0$	$C_6 = 0$	$C_6 = 1$
T_7	$\#T_6 = 1, \#T_7 = 0$	$C_7 = 0$	$C_7 = 1$
T_8	$\#T_7 = 1, \#T_8 = 0$	$C_8 = 0$	$C_8 = 1$

(a)

Initial	
$\#T_i = 0,\ i = 1 \dots 8$	
Target	Transition
T_1	$\#T_1 \leftarrow \#T_1 + 1$
T_2	$\#T_2 \leftarrow \#T_2 + 1$
$\vdots$	$\vdots$
T_8	$\#T_8 \leftarrow \#T_8 + 1$
Goal	
$\#T_1 = \#T_4 = \#T_5 = \#T_6 = \#T_7 = \#T_8 = 1$ $\#T_2 = \#T_3 = 3$	

(b)



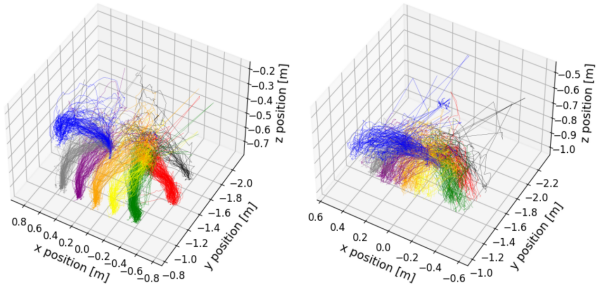
(c)

Fig. 4. High-level plan, represented by (a) target conditions, (b) state transitions and (c) a graphical representation, where $\#T_i$ is the number of completed reaching motions towards target T_i.

IV. Experiments and Results

In this section, the presented methods are evaluated and compared using data obtained from the publicly available HA4M dataset [6]. This dataset contains multi-modal data of different human subjects assembling an Epicyclic Gear Train (EGT). During the assembly of the EGT, the human subject has to reach for the different components, which are distributed over the worktable in a fixed configuration. From this dataset, twelve single-hand reaching motions, corresponding to eight different part locations, were extracted for each task demonstration. To successfully complete the assembly, these reaching motions had to be performed in accordance with a high-level plan, which is shown in Fig. 4. Fig. 4a describes the target conditions, which indicate whether reaching for a target T_k is in accordance with the high-level plan given the current state of the assembly, and provides their relation to the value of C_k. The factor C_k is used in the decision-making block of Fig. 3 to select the targets that require an intervention. The target conditions were derived from the dependencies between the components in the assembly of Fig. 1a. Fig. 4b describes the initial state, the goal state and the state transitions caused by each target. Here, the assumption is made that the target is known at the end of each reaching motion, such that the state transitions are deterministic. In Fig. 4c, the high-level plan is also visualized using a graph, where the arrows indicate the dependencies between the different targets, corresponding to the dependencies between the components in Fig. 1a (e.g., T_2, T_3 and T_4 should all be reached for, but only after T_1).

Separate evaluation experiments were performed on two datasets derived from the HA4M dataset: container picking and table picking (available at Zenodo: DOI 10.5281/zenodo.18363093). In the container picking dataset, containers were used to position the parts of the EGT at the start of the assembly, limiting the variations in the part locations between different demonstrations. In the table picking dataset, no containers were used to position the parts, causing the part locations to vary significantly. For both datasets, only the right-handed reaching motions were considered, resulting in 744 reaching motions with average duration 1.08s for container picking and 609 reaching motions with average duration 0.76s for table picking. To represent the reaching motions, only the position of the human's hand was used, as provided



(a) (b)

Fig. 5. Reaching motions from (a) the container picking dataset and (b) the table picking dataset, visualized in 3D. Reaching motions with the same target are visualized using the same color.

by the Azure Kinect camera and markerless skeleton tracking employed in the HA4M dataset. Note that raw measurement data were used without additional processing. Fig. 5 shows the resulting reaching motions, with average trajectory length 0.95m for container picking and 0.79m for table picking.

Since the part locations are not explicitly included in the HA4M dataset, the end-points of the reaching motions were used to approximate the target locations. For the container picking dataset, the target locations were estimated as the mean of all demonstrated reaching motion end-points corresponding to the same target, which approximates the constant container locations over all task demonstrations. Neighboring target locations were separated by distances between 0.21m (T_6 and T_7) and 0.41m (T_7 and T_8), with a mean separation of 0.28m. The target locations were distributed over the worktable in a U-shaped configuration around the participant, employing a workspace of approximately 1.7m wide and 0.8m long, as shown in Fig. 1. For the table picking dataset, the target locations were determined as the end-points of the reaching motions in the current task demonstration, since significant variations in the part locations occur between task demonstrations. On average, neighboring target locations were separated by distances between 0.12m (T_4 and T_5) and 0.25m (T_3 and T_4), with a mean separation of 0.18m. The target locations were also distributed over the worktable in a U-shaped configuration around the participant, employing a workspace of approximately 1.1m wide and 0.7m long.

The approaches explained in section III have a small number of tuning parameters remaining. The number of latent variables $m_{\max} = 6$ and the number of progress samples $N = 50$ for TraPPCA were derived from prior knowledge in literature [2], [7], [23]. The process noise $\sigma^2_{\text{process}} = 0.001$ was directly obtained from literature [2]. The additional variabilities for the motion models $\sigma_{\text{ADD,M}} = 0.1$m and the end-point predictions $\sigma_{\text{ADD,P}} = 0.1$m were experimentally tuned, but good initial estimates were obtained from application parameters, such as the size of the workspace and the separation of the target locations. The selected significance level $\alpha = 0.05$ is a common value in hypothesis testing. The priors for the models $P(M_j)$ and the targets $P(T_k)$ are assumed to be uniform, with $P(T_k) = 1/8$. For the unsupervised online learning approach, 5 random trajectories were used when initializing a new motion model. The random nature of this initialization can cause some small variations between different executions

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

of the datasets. Therefore, all results using unsupervised online learning shown in this section are the average over 5 executions of the datasets.

A. Evaluation of Intent Estimation

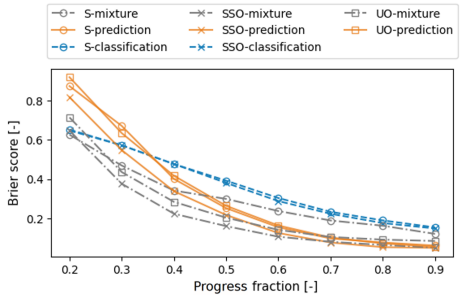
The methods section discussed three learning approaches for building a motion model database (III-D): a supervised offline (S), a semi-supervised online (SSO) and an unsupervised online (UO) approach. These learning approaches can each be combined with the intent estimation approaches of section III-C: classification, prediction and mixture intent estimation. The only exception is the combination of unsupervised online learning and classification intent estimation, which are not compatible. This section evaluates the intent estimation performance for these different combinations of learning and intent estimation approaches.

To ensure a fair comparison between the different learning and intent estimation approaches, each learning approach should receive (approximately) the same information before the performance assessment starts. Therefore, the first 10 task demonstrations of the dataset are not used to evaluate the intent estimation performance, but are only used as inputs to the learning approaches. The first 10 task demonstrations provide 82 and 67 right-handed reaching motions for container picking and table picking, respectively, both corresponding to 11% of the considered reaching motions. This limited number of initial demonstrations was selected in correspondence with the limited number of training demonstrations employed by related work in literature, as elaborated in section II. These initial demonstrations and their corresponding labels are used to construct the motion model database in the supervised offline learning approach and initialize the motion model database in the semi-supervised online learning approach. For both approaches, this results in a database consisting of 8 motion models, one corresponding to each predefined target. The same demonstrations without labels are also provided sequentially to the unsupervised online learning approach, constructing an initial motion model database.

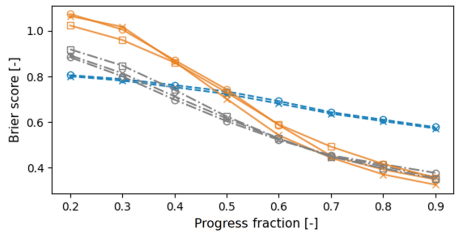
Then, the evaluation procedure iterates over the remaining task demonstrations and their consecutive reaching motions. For each reaching motion, a time-dependent trajectory $\boldsymbol{obs}_{t,r}$ is considered, with $r = 1 \dots R$ and R the number of considered reaching motions during evaluation. This trajectory is first used by the different intent estimation approaches to predict the target probabilities $P(T_k|\boldsymbol{obs}_{t,r})$ at each of the considered progress fractions (20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%). Afterwards, the complete trajectory is used to update the motion model database of the semi-supervised and unsupervised online learning approach, such that these updated databases can be used to predict the intent of the next reaching motion.

To measure the intent estimation performance for the different combinations of learning and intent estimation approaches, the Brier score [31] is calculated at each of the considered progress fractions using following equation:

$$\text{BS} = \frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{N_T+1}\left(P(T_k|\boldsymbol{obs}_{t,r}) - E_{rk}\right)^2, \qquad (25)$$



(a) Container picking



(b) Table picking

Fig. 6. Brier score for different combinations of learning and intent estimation approaches, evaluated at the considered progress fractions (20%, 30%, 40%, 50%, 60%, 70%, 80% and 90%).

with N_T as the number of targets and $P(T_k|\boldsymbol{obs}_{t,r})$ as the target probabilities of section III-E. The parameter E_{rk} represents the ground truth target of $\boldsymbol{obs}_{t,r}$. It is 1 if the considered target T_k is the ground truth target and is 0 otherwise. This results in a score between 0 and 2 for the different approaches at each of the considered progress fractions (Fig. 6), with a lower Brier score indicating a better performance.

For the container picking dataset, Fig. 6a shows that the intent estimation performance at the start of the reaching motion is mainly determined by the employed intent estimation approach. Initially, after observing 20% of the reaching motion, the worst performance is obtained for prediction intent estimation. At 20%, only limited information is observed about the ongoing trajectory and limited differentiation is present between the considered motion models, resulting in high uncertainty about the performed motion type. Meanwhile, prediction intent estimation requires the motion model that best corresponds to the observed partial trajectory to be selected. Due to the high uncertainty early in the reaching motion, this choice of best corresponding motion model is not reliable, which explains the worse intent estimation performance.

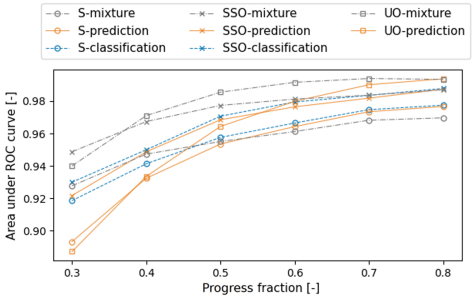
The performance using classification intent estimation only slowly improves with the observed progress fraction. Therefore, once 40% of the motion is observed, classification intent estimation provides the worst performance, regardless of the employed learning approach. Meanwhile, for mixture intent estimation, the employed learning approach starts to influence the performance. When supervised learning is used (S-mixture), a worse performance is obtained towards the end of the reaching motion, approaching the performance obtained using classification intent estimation. However, when semi-supervised online learning (SSO-mixture) is used, the best performance over the whole reaching motion is obtained. Only towards the end of the reaching motion, a similar performance is obtained for prediction intent estimation.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

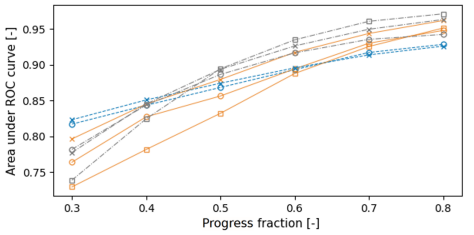
This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

For the table picking dataset, Fig. 6b shows that the intent estimation performance throughout the entire reaching motion is mainly determined by the employed intent estimation approach. At the start of the reaching motion, the worst performance is again obtained for prediction intent estimation, which has the same explanation as previously provided for the container picking dataset. However, additional uncertainty is introduced by not using containers to position the parts. This generally decreases the obtained performance, but also causes the performance for prediction intent estimation to remain the worst until a large part (50%) of the motion is observed.

The performance using classification intent estimation again only slowly improves with the observed progress fraction. Therefore, once 50% of the motion is observed, classification intent estimation provides the worst performance. Meanwhile, since the uncertainty about the performed motion type is limited at the end of the reaching motion, mixture intent estimation starts to favor specific motion models. This causes mixture intent estimation to converge towards prediction intent estimation at the end of the reaching motion.



(a) Container picking



(b) Table picking

Fig. 7. Area under ROC curve for different combinations of learning and intent estimation approaches, evaluated at the considered progress fractions (30%, 40%, 50%, 60%, 70% and 80%).

B. Evaluation of Intent-Based Decision Making

In this section, the performance of the intent-based decision-making approach using error probability P_{error} (III-E) is evaluated for the different learning and intent estimation approaches. To perform this evaluation, P_{error} is first calculated for each probability distribution $P(T_k|\boldsymbol{obs}_{t,r})$ obtained in section IV-A and different assembly states in accordance with the high-level plan. These assembly states include the actual assembly state according to the HA4M dataset, but also all past and possible future assembly states, introducing situations in which the current reaching target deviates from the high-level plan. Next, these error probabilities are compared with a series of thresholds between 0 and 1, making a decision for each threshold. These decisions are used to determine the true positive rate (TPR) and false positive rate (FPR) for each threshold at the considered progress fractions (30%, 40%, 50%, 60%, 70% and 80%), which is visualized using receiver operating characteristic (ROC) curves (Fig. 3 in Supplemental Material). Finally, the area under the ROC curve (AUC) is calculated for each combination of learning and intent estimation approaches in function of the observed progress fraction (Fig. 7). This provides a measure of the intent-based decision-making performance located between 0 and 1, with a higher value indicating a better performance.

For the container picking dataset, Fig. 7a shows that the best overall decision-making performance is achieved by combining mixture intent estimation with unsupervised or semi-supervised online learning. Initially, when 30% of the reaching motion is observed, the best performance for each learning approach is obtained using mixture intent estimation, while the worst performance is obtained using prediction intent estimation. The latter is again explained by the high uncertainty at the start of the reaching motion, as previously mentioned in section IV-A. Meanwhile, the learning approach that provides the best performance at the start is semi-supervised online learning. Towards the end of the reaching motion (70-80%), the performance is mainly determined by the employed learning approach. The best performance is obtained using unsupervised online learning, while the worst performance is obtained using supervised learning.

For the table picking dataset, Fig. 7b shows a general decrease in performance compared to the container picking dataset. This is caused by the additional uncertainty introduced by not using containers to position the parts. Initially, when 30% of the reaching motion is observed, the worst performance is obtained for all approaches using unsupervised online learning, while the best performance is obtained for all approaches using classification intent estimation. However, the performance using classification intent estimation only slowly improves with the observed progress fraction, which is consistent with the observations of section IV-A. Therefore, once 70% of the reaching motion is observed, classification intent estimation provides the worst performance. Meanwhile, once 50% of the reaching motion is observed, the best performance is obtained by combining mixture intent estimation with unsupervised or semi-supervised online learning.

C. Evaluation of Online Learning

This section evaluates how the proposed semi-supervised and unsupervised online learning approaches affect both the adaptability of the motion model database and the intent-based decision-making performance. It examines how these effects evolve as more reaching motions are observed, highlighting how learning behavior and intent-based decision making change when more of the encountered reaching motions were previously observed.

First, the semi-supervised and unsupervised online learning approaches sequentially observed the reaching motions in the test data, storing a separate, updated motion model database after each reaching motion. Then, for each database, the corresponding learning approach determined how many motions in

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

the test data would still be classified as "other" when observed next. The evolution of this "other" count over the number of observed reaching motions quantifies the ability of the learning approach to capture previously unseen motion types, with a larger reduction in "other" count indicating a more effective integration of new motion types. Also, for each database, the performance of the intent-based decision-making approach using P_{error} (III-E) was evaluated over the test data using each of the proposed intent estimation approaches, quantifying the performance using AUC. Fig. 8 shows the results for container picking and 70% prediction fraction.

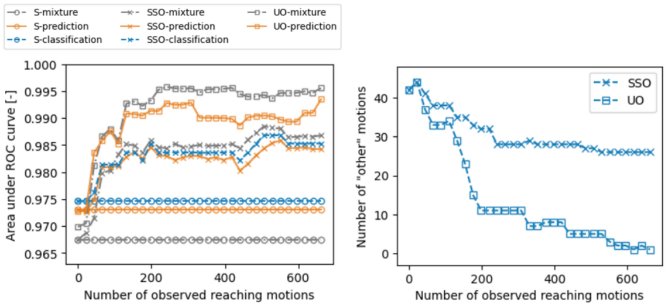
For the unsupervised online learning approach, the number of "other" motions decreased from 42 before observing the test data to 1 after (Fig. 8b), demonstrating the approach's ability to effectively capture new motion types. By updating the motion model database, this learning approach also provided the largest performance improvement in Fig. 8a. However, the cost of these improvements is an increased number of motion models, with the database containing 19 models after observing the test data. Therefore, future work should investigate approaches to periodically reduce the number of models in the database (e.g., removing motions, reclustering motions).

For the semi-supervised online learning approach, the number of "other" motions decreased from 42 before observing the test data to 26 after (Fig. 8b), which demonstrates that the approach is less effective at capturing motion types that differ substantially from those already encoded in the motion model database. Consequently, this approach is less likely to include irrelevant or erroneous motions in the database, but risks excluding new, relevant motion types. In Fig. 8a, by updating the motion model database, this learning approach provides a performance improvement compared to supervised offline learning, but its performance remains below the unsupervised online learning approach.

For both online learning approaches, the largest improvements were observed early in the learning process (e.g., first 250 motions in Fig. 8). However, in Fig. 8a, the performance improvements are small (max. 0.026 for UO-mixture). After the initial training phase, the motion model databases already represented most of the reaching motions in the test data, leaving limited room for improvement. This is also demonstrated by the limited "other" motions after the initial training phase, corresponding to 6.3% of the motions for both online learning approaches. Nevertheless, the above results demonstrate the benefits and drawbacks of these online learning approaches, warranting further testing on more challenging cases.

D. Comparison of Decision-Making Approaches

This section provides a comparison between the two decision-making approaches of section III-E: (1) decision making using error probability P_{error} (probabilistic approach) and (2) decision making using the most probable target (intended target approach). In the probabilistic approach, to decide whether an intervention in the human's task execution is required, the error probability P_{error} is calculated and compared with an intervention threshold between 0 and 1. As mentioned before, an advantage of this approach is that the decision-making process can be adapted to different situations through



(a) (b)

Fig. 8. Evolution of (a) intent-based decision-making performance (area under ROC curve) and (b) "other" classification count over the number of observed reaching motions for different learning (and intent estimation) approaches on the container picking dataset (70% prediction fraction).

the selection of the employed threshold. The optimal value for this threshold is found by maximizing the expected utility

$$\text{EU} = \text{TPR} \cdot u_{\text{TP}} + \text{FPR} \cdot u_{\text{FP}}, \tag{26}$$

given the utilities u_{TP} and u_{FP} of a true positive and a false positive decision, respectively. These utilities are also related to the slope of the iso-utility lines in ROC space:

$$u_{\text{TP}} = \frac{1}{1+\text{slope}}, \quad u_{\text{FP}} = \frac{-\text{slope}}{1+\text{slope}}, \tag{27}$$

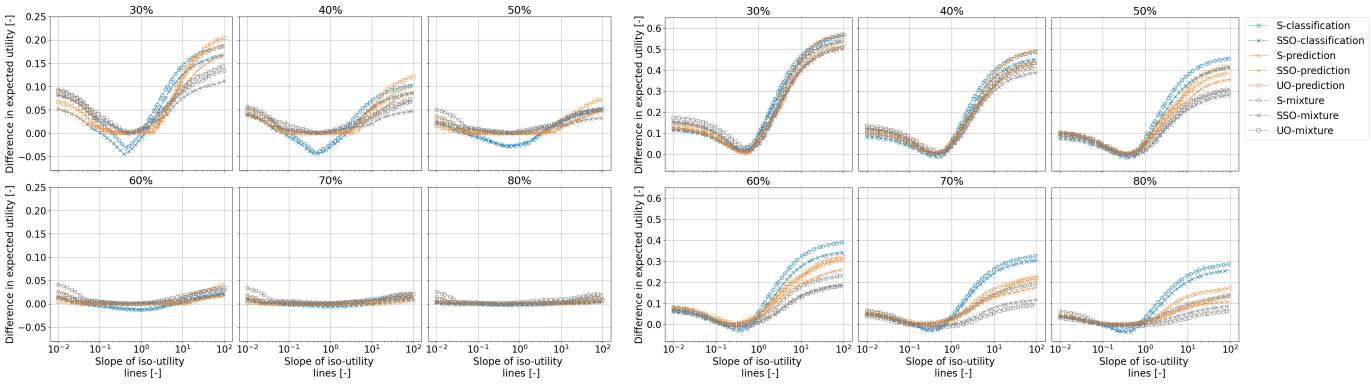
given the assumption that $|u_{\text{TP}}| + |u_{\text{FP}}| = 1$. The optimal threshold for the current slope is then found at the point where the iso-utility line is tangent to the ROC curve.

In the intended target approach, to decide whether an intervention is required, the most probable target is selected as the intended target and compared with the high-level plan. This decision-making process cannot be adapted to different situations and will always provide the same decision for a given probability distribution $P(T_k|\boldsymbol{obs}_{t,r})$ and a certain high-level plan. Therefore, this approach corresponds to a single point in ROC space (Fig. 3 in Supplemental Material).

The difference in expected utility between the probabilistic approach and the intended target approach can then be expressed in function of the slope of the iso-utility lines (Fig. 9). Here, small slopes indicate that true positives are considered more important than false positives, while large slopes indicate that true positives are considered less important than false positives. For the container picking dataset, Fig. 9a shows that, whenever classification intent estimation is used, the probabilistic approach results in a lower expected utility for a significant part of the considered slopes, which indicates that the intended target approach is better suited in those situations. However, when prediction intent estimation or mixture intent estimation is used, the probabilistic approach results in a higher or similar expected utility for all considered slopes. Note that the intended target approach performs best for a specific value of the slope, which depends on the application. When deviating from this optimal value, improvements are obtained by employing the probabilistic approach, demonstrating its adaptability to different situations. The largest improvements are obtained at the start of the reaching motion, since the probabilistic approach is better at handling large uncertainties. Towards the end of the reaching motion, the

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918



(a) Container picking

(b) Table picking

Fig. 9. Comparison of the decision-making approaches for different progress fractions (30%, 40%, 50%, 60%, 70% and 80%) and different combinations of learning and intent estimation approaches. The figures show the difference in expected utility $\mathrm{EU}_{\mathrm{prob}} - \mathrm{EU}_{\mathrm{target}}$ between decision making using P_{error} ($\mathrm{EU}_{\mathrm{prob}}$) and decision making using the most probable target ($\mathrm{EU}_{\mathrm{target}}$) as a function of the slope of the iso-utility lines, with expected utility calculated according to equations (26) and (27).

uncertainty decreases, such that the expected utilities of both approaches converge for all considered slopes.

For the table picking dataset, Fig. 9b shows that, at the start of the reaching motion, the probabilistic approach results in a higher expected utility for all considered slopes and all combinations of learning and intent estimation approaches. This is caused by the additional uncertainty introduced by not using containers to position the parts, with most uncertainty being present at the start of the reaching motion. As the observed progress fraction increases, the difference in expected utility between both decision-making approaches decreases. However, unlike the container picking dataset, the expected utilities of both approaches do not converge within the considered progress fractions. When classification intent estimation is used, the probabilistic approach results in slightly lower expected utilities around a slope of 0.4, which corresponds to the optimal slope of the intended target approach. However, once 60% of the reaching motion is observed, classification intent estimation benefits most from the probabilistic approach at large slopes (slope > 1). When prediction intent estimation or mixture intent estimation is used, the probabilistic approach results in a higher or similar expected utility for all considered slopes. Most improvement is again obtained when deviating from the optimal slope of the intended target approach.

V. Discussion and Conclusion

This work presented intent-based decision-making approaches for operator assistance in manipulation tasks, which are based on three main contributions. First, three novel approaches for reaching motion intent estimation were introduced: classification, prediction and mixture intent estimation. Here, the assumption was made that the human's intent is solely determined by the target of the reaching motion, only considering the shape of the motion as a means to predict this target. However, as an extension, the classification intent estimation approach would also be able to distinguish between different motion types with the same target based on the shape of the motion, given that the same distinction was also made when learning the employed motion model database. Second, two novel approaches were introduced for learning a motion model database, which are based on semi-supervised and unsupervised online learning. An advantage of these online learning approaches is that new motion styles observed during execution can be encoded in the motion model database. However, future work should investigate how these online learning approaches cope with non-stationary problems and how their performance in these conditions can be improved. Third, a decision-making approach for human operator assistance in manipulation tasks was introduced, which directly considers the target probabilities obtained from intent estimation. An advantage of this approach is that the decision-making process can be adapted to different situations based on a threshold. This threshold can be selected by maximizing the expected utility given the utilities of a true positive and a false positive decision. However, the selection of these utilities is dependent on the application and is outside the scope of this work.

The above contributions were validated on two datasets derived from the publicly available HA4M dataset: container picking and table picking. First, the intent estimation performance and the intent-based decision-making performance were compared for the above learning and intent estimation approaches, demonstrating the advantages and disadvantages of the different approaches. Then, the proposed online learning approaches were compared by examining how learning behavior and intent-based decision-making performance change as more reaching motions are observed. Finally, the proposed decision-making approach was compared with a baseline approach using only the most probable target. This showed that the proposed approach is more adaptable to different situations and is better at handling large uncertainties. Meanwhile, the baseline approach performs best for a specific choice of utilities, only exceeding the performance of the proposed approach for classification intent estimation and utilities close to the optimal choice for the baseline approach.

In the above experiments, only raw measurements of hand positions were used to represent the reaching motions, demonstrating that the presented approaches can deal with limited information and noisy data. Nonetheless, the presented approaches can also be extended to account for additional signals, such as the position of certain key points on the arm

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Human-Machine Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/THMS.2026.3675918

(e.g., wrist, elbow, shoulder) and torso and the orientation of the hand, which provide additional information, especially in more complex applications. To properly learn the correlations between a larger number of signals, TraPPCA would however require more data. Therefore, the supervised offline and semi-supervised online learning approach would require more manually-labelled training data during the offline training phase and the unsupervised online learning approach would require more online executions to construct a proper motion model database.

Furthermore, the presented approaches assumed reaching motions with a constant intent. The unsupervised online learning approach would be able to include reaching motions with changing intent into the motion model database, but due to the large variety of possible intent changes (e.g., between different targets, at different progress fractions), many reaching motions must be observed to capture all possible intent changes. Therefore, when a reaching motion with changing intent is observed, it will most likely be classified as "other" by the proposed intent estimation approaches, causing no intervention in the human's task execution. This behavior is reasonable, since intent changes in human reaching motions generally leave insufficient time to re-estimate the intended target and provide an early intervention.

Compared to traditional approaches for operator assistance in manipulation tasks (e.g., Pick to Light), which guide the human operator through the task using feedforward instructions (e.g., lights near each target), the proposed intent-based decision-making approaches provide several advantages: more freedom for the operator, less additional hardware and a more flexible shop floor (e.g., the human operator can give priority to urgent subtasks without reprogramming of the system).

In future work, online experiments will be conducted on a real workstation with 3D part storage, introducing additional complexity. Online experiments also imply that no prior segmentation of the reaching motions can be performed, such that an onset detection approach, similarly as described by Callens et al. [2], is required to detect the start of each reaching motion. Additionally, the detection of "other" motions will be tested more rigorously, which was not the focus of the current experiments due to insufficient "other" motions in the HA4M dataset. Finally, future work will investigate more specifically the application of the presented approaches in HRC.

REFERENCES

[1] D. Nicolis, A. M. Zanchettin, and P. Rocco, "Human intention estimation based on neural networks for enhanced collaboration with robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 1326 – 1333.

[2] T. Callens, A. van der Have, S. Van Rossom, J. De Schutter, and E. Aertbeliën, "A framework for recognition and prediction of human motions in human-robot collaboration using probabilistic motion models," *IEEE Robot. Automat. Letters (RA-L) paper presented at 2020 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 5, no. 4, pp. 5151 – 5158, 2020.

[3] A. Gellert, S.-A. Precup, A. Matei, B.-C. Pirvu, and C.-B. Zamfirescu, "Real-time assembly support system with hidden markov model and hybrid extensions," *Mathematics*, vol. 10, no. 15, 2022, Art. no. 2725.

[4] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team," in *2nd ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, 2007, pp. 1 – 8.

[5] G. Ganesh, A. Takagi, R. Osu, T. Yoshioka, M. Kawato, and E. Burdet, "Two is better than one: Physical interactions improve motor performance in humans," *Sci. Rep.*, vol. 4, 2014, Art. no. 3824.

[6] G. Cicirelli, R. Marani, L. Romeo, M. García Domínguez, J. Heras, A. G. Perri, and T. D'Orazio, "The HA4M dataset: Multi-modal monitoring of an assembly task for human action recognition in manufacturing," *Sci. Data*, vol. 9, 2022, Art. no. 745.

[7] E. Aertbeliën and J. De Schutter, "Learning a predictive model of human gait for the control of a lower-limb exoskeleton," in *5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatron.*, 2014, pp. 520 – 525.

[8] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688 – 1703, 1985.

[9] C. T. Landi, Y. Cheng, F. Ferraguti, M. Bonfe, C. Secchi, and M. Tomizuka, "Prediction of human arm target for robot reaching movements," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 5950 – 5957.

[10] A. M. Zanchettin and P. Rocco, "Probabilistic inference of human arm reaching target for effective human-robot collaboration," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 6595 – 6600.

[11] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *IEEE Int. Conf. Robot. Automat.*, 2015, pp. 885 – 892.

[12] O. S. Oguz, Z. Zhou, and D. Wollherr, "A hybrid framework for understanding and predicting human reaching motions," *Frontiers Robot. AI*, vol. 5, 2018, Art. no. 27.

[13] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.

[14] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2013, pp. 299 – 306.

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

[16] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *IEEE Int. Conf. Robot. Automat. (ICRA)*, 2015, pp. 6175 – 6182.

[17] R. Luo and D. Berenson, "A framework for unsupervised online human reaching motion recognition and early prediction," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2015, pp. 2426 – 2433.

[18] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Auton. robots*, vol. 42, no. 3, pp. 631 – 648, 2018.

[19] P. Song, P. Li, E. Aertbeliën, and R. Detry, "Robot Trajectron: Trajectory prediction-based shared control for robot manipulation," in *IEEE Int. Conf. Robot. Automat. (ICRA)*, 2024, pp. 5585 – 5591.

[20] S. Wang, J. Yin, J. Cao, M. Tang, H. Wang, and Y. Zhang, "ABG-NAS: Adaptive bayesian genetic neural architecture search for graph representation learning," *Knowl.-Based Syst.*, vol. 328, p. 114235, 2025.

[21] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43 – 49, 1978.

[22] T. Callens, J. De Schutter (sup.), and E. Aertbeliën (cosup.), "Assistance strategies for quasi-passive and active occupational exoskeletons based on intention estimation," PhD thesis, KU Leuven, 2023.

[23] C. Vergara, J. De Schutter, and E. Aertbeliën, "Learning robust manipulation tasks involving contact using trajectory parameterized probabilistic principal component analysis," in *2020 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 8336 – 8343.

[24] Y. Li, S. Wang, S. Xu, and J. Yin, "Trustworthy semi-supervised anomaly detection for online-to-offline logistics business in merchant identification," *CAAI Trans. Intell. Technol.*, vol. 9, no. 3, pp. 544 – 556, 2024.

[25] P. Zarchan and H. Musoff, *Fundamentals of Kalman filtering: a practical approach*, 2nd ed. AIAA, 2005.

[26] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. John Wiley & Sons, 2004.

[27] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Stat. Assoc.*, vol. 56, no. 293, pp. 52 – 64, 1961.

[28] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*, 1st ed. Springer International Publishing AG, 2023.

[29] Y. Bengio, A. Courville, and I. Goodfellow, *Deep learning*, ser. Adaptive Computation and Machine Learning. MIT Press, 2016.

[30] S. Calinon, "A tutorial on task-parameterized movement learning and retrieval," *Intell. Service Robot.*, vol. 9, no. 1, pp. 1 – 29, 2016.

[31] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Mon. Weather Rev.*, vol. 78, no. 1, pp. 1 – 3, 1950.

© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.