

Process Mining Over Sensor Data: Goal Recognition for Powered Transhumeral Prostheses

Zihang Su 

The University of Melbourne
zihangs@student.unimelb.edu.au

Artem Polyvyanyy 

The University of Melbourne
artem.polyvyanyy@unimelb.edu.au

Nir Lipovetzky 

The University of Melbourne
nir.lipovetzky@unimelb.edu.au

Nick van Beest 

Data61 | CSIRO
nick.vanbeest@data61.csiro.au

Tianshi Yu 

The University of Melbourne
tianshiy@student.unimelb.edu.au

Ying Tan 

The University of Melbourne
yingt@unimelb.edu.au

Sebastian Sardiña 

RMIT University
sebastian.sardina@rmit.edu.au

Alireza Mohammadi 

The University of Melbourne
alireza.mohammadi@unimelb.edu.au

Denny Oetomo 

The University of Melbourne
doetomo@unimelb.edu.au

Wednesday 12th March, 2025

Abstract

Process mining (PM)-based goal recognition (GR) techniques, which infer goals or targets based on sequences of observed actions, have shown efficacy in real-world engineering applications. This study explores the applicability of PM-based GR in identifying target poses for users employing powered transhumeral prosthetics. These prosthetics are designed to restore missing anatomical segments below the shoulder, including the hand. In this article, we aim to apply the GR techniques to identify the intended movements of users, enabling the motors on the powered transhumeral prosthesis to execute the desired motions precisely. In this way, a powered transhumeral prosthesis can assist individuals with disabilities in completing movement tasks. PM-based GR techniques were initially designed to infer goals from sequences of observed actions, where discrete event names represent actions. However, the electromyography electrodes and kinematic sensors on powered transhumeral prosthetic devices register sequences of continuous, real-valued data measurements. Therefore, we rely on methods to transform sensor data into discrete events and integrate these methods with the PM-based GR system to develop target pose recognition approaches. Two data transformation approaches are introduced. The first approach relies on the clustering of data measurements collected before the target pose is reached (the clustering approach). The second approach uses the time series of measurements collected while the dynamic user movement to perform linear discriminant analysis (LDA) classification and identify discrete events (the dynamic LDA approach). These methods are evaluated through offline and human-in-the-loop (online) experiments and compared with established techniques, such as static LDA, an LDA classification based on data collected at static target poses, and GR approaches based on neural networks. Real-time human-in-the-loop experiments further validate the effectiveness of the proposed methods, demonstrating that PM-based GR using the dynamic LDA classifier achieves superior F_1 score and balanced accuracy compared to state-of-the-art techniques.

Keywords: Goal recognition, transhumeral prostheses, process mining, human-in-the-loop

1 Introduction

This research investigates the feasibility and effectiveness of applying process mining (PM)-based goal recognition (GR) techniques to enhance the performance of *powered transhumeral prosthetics*. A transhumeral prosthesis is designed to replace the missing anatomical parts of an arm below the patient’s shoulder. A powered prosthesis is equipped with sensors and motors designed to recognize the user’s intentions and provide assistance. For instance, it can autonomously steer the artificial limb toward the target prosthetic pose, supporting the patient in replicating functions of a natural arm.¹

The state-of-the-art techniques for inferring target poses, such as Linear Discriminant Analysis (LDA) classifier [38] and deep learning-based classifier [13], have limitations. The LDA classifier distinguishes target poses based on individual data points rather than considering historical sequences of data points, thereby failing to utilize historical information for inference. In turn, the deep learning-based classifier grounded in the long short-term memory (LSTM) network requires a large training dataset. Consequently, it is often challenging to instruct a subject to repeat the same movement numerous times to generate sufficient training data. Additionally, the signal patterns vary significantly between individuals [32], making it impractical to pool data from multiple subjects for training a single model.

GR is a promising technique for assisting powered transhumeral prosthetic devices, as it aims to infer the final (goal) state of an agent based on a sequence of their so far observed actions. GR techniques have been successfully applied in a wide range of scenarios, including robotics [8, 17], autonomous driving [1, 3], and human-machine interaction [28]. This study investigates the possibility of using sensor data collected by transhumeral prostheses to infer users’ intended poses and thus support the users in reaching these poses. Once the goal is identified, it becomes feasible to plan a smooth joint movement trajectory connecting the current position to the target pose. This fosters better cooperation between the user and the prosthesis, leading to more natural movements and, consequently, improving the efficiency of prosthetic use for individuals with upper limb disabilities [19, 37, 38]. Accurately identifying the goal is crucial for prosthesis movement control, as failure to achieve it can result in inefficient task execution, user frustration, and potential device abandonment [10].

Despite the potential of GR, popular GR techniques often depend on human-defined domain models. While these models perform well in synthetic domains, defining them for complex real-world scenarios, such as the prosthetic domain, is challenging. In contrast, PM-based GR automatically learns models from historical data, eliminating the need for human-crafted models and making it well-suited for real-world applications [30]. The challenge of using PM-based GR techniques in the development of powered transhumeral prostheses lies in preprocessing sensor-collected data to align with the input format required by process discovery techniques. An input to a PM-based GR technique consists of a sequence of actions, each captured as a discrete event, for example “a movement from point A to point B” or “a completion of a task.” These actions are described using discrete names or labels. In contrast, a powered transhumeral prosthetic device registers high-frequency, continuous, real-valued sensor

¹In this article, we use the terms “target pose” and “goal” interchangeably.

signals, such as 0.05mV for electromyography signals, which measure the electric potential generated by muscle cells, or 4.2-precision degrees for kinematic signals, which measure the angle of the elbow joint. To implement a PM-based GR system over such sensor measurements, we transform these measurements into discrete events. In our previous work [32], the clustering-based data discretization method did not incorporate the target pose information associated with each trajectory, and it required access to the complete dataset. Consequently, while we demonstrated that PM-based GR techniques could address key challenges in prosthesis development, the proposed method was less suitable for practical, real-world applications. In this article, we introduce a linear discriminant analysis (LDA) classifier that can be pre-trained using historical data. This approach enables the system to learn clustering criteria in advance, allowing it to operate in real-time and interact dynamically with patients.

We evaluated our approach through two experimental settings: an offline experiment and a human-in-the-loop (HITL) experiment. The HITL experiments introduced in this paper are specifically designed to validate the method’s performance in real-world scenarios, enabling humans to interact and collaborate with the prosthesis to complete a task. In the offline experiment, we used an existing dataset collected for the development of powered transhumeral prostheses [32]. The dataset comprises data from ten non-disabled subjects. Each subject was instructed to perform forward-reaching tasks involving three distinct elbow poses while the kinematic and electromyography sensor data was collected. After data collection, a portion of the data was used to train our GR system, while the remaining data was used as a testing set to evaluate the accuracy of the goal inference. In the human-in-the-loop (HITL) experiment, we invited another six non-disabled subjects to participate, instructing them to grasp and relocate clothespins, as documented in [14, 19]. In the HITL experiments, we used the same data collection method as in the offline experiment, and all the collected data was used to train the GR system. To evaluate the quality of our GR system inferences, we provided the trained system to the participants, allowing them to interact with the prosthesis guided by our goal inferences in real-time (in a VR environment) to complete the clothespin relocation tasks. Subsequently, we assessed the performance based on how effectively the subjects could use the VR prosthesis device to execute the tasks. During the online testing², a subject may produce inputs that are sparsely covered by the training data due to the reactions of the subject to the real-time prosthesis movement controlled by GR inferences. Thus, the HITL experiments are crucial and serve as a robust measure for testing the effectiveness of the developed GR-enabled powered transhumeral prostheses.

We compare the results against existing state-of-the-art baselines. The existing LDA baseline [38] is trained using data collected only at target poses, without considering trajectories leading to them, referred to as the “static” LDA classifier. To improve upon this, we modify the training data to include data points along the entire trajectory, which we refer to as the “dynamic” LDA method. Experimental results show that the dynamic LDA outperforms the static LDA classifier. Another baseline we used

²The human-in-the-loop experiment and the online experiment both refer to experiments conducted in a real-time setting. In these experiments, the subject interacts with the prosthetic device to collaborate and complete tasks. Therefore, in this paper, the terms “human-in-the-loop experiment,” “online experiment,” and “real-time experiment” are used interchangeably and share the same meaning.

is the long short-term memory (LSTM) neural network-based target pose recognition approach [13]. Thus, we used three baselines: static LDA, dynamic LDA, and LSTM.

The offline experiments revealed that two techniques stand out as the most effective: dynamic LDA and the PM-based GR technique with the LDA classifier used for data discretization (our new approach). Specifically, our approach achieved the highest balanced accuracy, while dynamic LDA achieved the highest F_1 score. Subsequently, human-in-the-loop (HITL) experiments assess the real-time practicality of the two top-performing techniques selected from the offline experiments. In these online experiments, our PM-based GR with LDA classifier achieved significantly higher F_1 scores and balanced accuracy than the dynamic LDA approach.

This article is an extended version of our conference paper [32], which made the following contributions:

1. An extension of the data-driven approach for GR grounded in process mining techniques [26, 30] to scenarios where multi-dimensional, real-valued, continuous measurements characterize the behavior of the observed agent.
2. The results of an evaluation based on open-source implementations of several state-of-the-art GR systems³, including the PM-based system, over a publicly available dataset⁴ in the domain of transhumeral prostheses which confirm that the PM-based system achieves significantly superior performance.

Our previous work relied on clustering to handle multi-dimensional, real-valued, continuous measurements, which limited its applicability in real-time settings due to the need to access the complete dataset. In this article, we aim to improve the method by using a pre-trained classifier, enabling online application through real-time interaction with patients performing tasks. Human-in-the-loop (HITL) experiments are conducted to evaluate the technique's performance in real-world task interactions.

Concretely, this article provides these additional contributions:

1. An enhanced version of the state-of-the-art LDA approach to goal inference, tailored for scenarios where trajectories of signals are available. Specifically, a novel dynamic LDA method is introduced as an extension of the classic LDA method to handle movement trajectories. Offline experiments demonstrate that dynamic LDA surpasses the static approach in performance.
2. An improved iteration of the PM-based GR approach, incorporating an LDA classifier to convert signals into event labels. Our offline experiments show that this enhanced approach achieves superior performance compared to the method utilized in the conference paper and comparable performance with the dynamic LDA approach.
3. Human-in-the-loop (HITL) experiments to assess the two best-performing approaches identified in the offline setting: the dynamic LDA technique and the PM-based GR with LDA classifier. The results indicate that the PM-based technique outperforms dynamic LDA, aiding users in achieving their goals more rapidly. This underscores its potential to facilitate natural movements for users.

³<https://doi.org/10.26188/24131493>

⁴<https://doi.org/10.26188/23294693>

All experimental datasets⁵ and implemented approaches⁶ have been made publicly available. The next section provides the necessary background on PM-based GR and our transhumeral prosthetic studies and experiments. Then, Section 3 discusses related work. Section 4 presents our approaches to goal inference, while Section 5 presents the conducted experiments and their results. Section 6 discusses the limitations of the current techniques and future work aiming to address these limitations. Finally, Section 7 states the concluding remarks.

2 Background

This section presents background on the PM-based GR (Section 2.1) and the conducted prosthetic experiments, including the offline and HITL experimental settings (Section 2.2).

2.1 Process Mining-Based Goal Recognition

Given a collection of candidate goals and observations of actions performed by an agent in an environment, a solution to the goal recognition (GR) problem suggests the true goal the agent strives to achieve [23]. Process mining (PM) techniques were recently used to implement a system for solving the GR problem [26, 30]. We refer to this system as the PM-based GR system.

PM-based GR system uses *process discovery* [18] to construct process models from the sequences of historical actions the agent used to achieve the candidate goals in the past. Specifically, each model is constructed from the historical observations of how the agent reached a certain candidate goal and, thus, represents the standard behavior for achieving the goal. In process mining, an executed action is referred to as an *event*, while a sequence of executed actions is called a *trace*. With the process models at hand, one per possible goal, the PM-based GR system uses *conformance checking* [33] to align a newly observed action sequence—a trace—with each discovered model. Finally, the commonalities and discrepancies found between the observed trace and each model induce a *probability distribution over the candidate goals* representing the likelihood of each goal being the one the agent is pursuing. As the PM-based GR technique comprises constructing process models from historical data (collections of observed traces toward the various goals, called *event logs*) and conformance checking between newly observed traces and the constructed models, it is a data-driven GR technique.

2.2 Prosthetic Experiments

This paper conducts a two-step study that consists of (i) offline experiments for algorithm development and (ii) human-in-the-loop (HITL) prosthesis movement control experiments employing the developed algorithm. This two-step approach is commonly employed in the literature of powered prostheses [10]. We first utilize pre-collected

⁵<https://doi.org/10.26188/25488130>

⁶<https://doi.org/10.26188/25487290>

body movement and muscle activity datasets from non-disabled human subjects to develop an accurate and robust GR algorithm based on offline experiments. Then, we deploy the algorithm to control the prosthetic device in real-time based on the inputs generated by human users that are impacted by the HITL control mechanisms.

The distinction between offline and HITL experiments is whether the subjects interact with the powered prosthesis. In offline experiments, measurements are taken as non-disabled subjects reach goals with their intact limb without interacting with the prosthesis. This dataset is collected to inform the development of the GR system. In HITL experiments, real-time measurements are used by the developed GR system to recognize the goal. The motors then drive the prosthesis to the recognized goal (pose), which closely aligns with the intended real-life use of the prosthetic device. In a HITL experiment, the subject interacts with the prosthesis by experiencing its movement and adjusting their joint and muscle movements accordingly. These “interactions” between the prosthesis and the user can lead to inputs to the GR system that are substantially different from those captured in the offline dataset. HITL experiments thus support testing the robustness of the developed GR system. Additionally, doubts exist in the literature regarding the correlation between offline and HITL performance [12, 21, 24], highlighting the need for HITL experiments to justify the efficacy of the developed powered transhumeral prostheses.

2.2.1 Offline Experiments

The dataset for offline experiments was collected as non-disabled subjects extended their intact upper limbs forward to reach three goal elbow positions repeated at three shoulder flexion/extension poses. The goals (target elbow poses) are denoted as $T1$, $T2$, and $T3$ in Fig. 1a, which illustrates the side-view schematic of the upper limb. The dataset captures the above-elbow joint movements and muscle activities through motion trackers and surface electromyography (sEMG) sensors, respectively, with 12 joint kinematic movement features and 35 sEMG features extracted at a rate of 10 Hz (the measurements were taken every 0.1 seconds). The motion trackers determine their position and rotation using infrared (IR) distance sensors and embedded inertial measurement units (IMUs), enabling the resolution of joint kinematics. The sEMG sensors are adhered to the skin above the target muscles, with their electrodes detecting the electrical signals generated by the muscles during muscle contraction. The process of extracting the features has been described in detail in previous work [38].

The sensor placements and the virtual avatar in VR are shown in Figs. 1b and 1c, respectively. The virtual avatar serves as the visual representation of the user’s presence in VR, enabling interaction with the virtual environment, such as reaching the virtual target. To capture the residual limb joint kinematics, three HTC VIVE trackers were strategically positioned at the upper arm (UA), shoulder acromion (SA), and trunk (TR). The displacement and velocity of the six degrees of freedom (DoF) shoulder and trunk movements were extracted as features. An additional tracker was placed on the forearm (FA) to acquire the elbow joint kinematics. A controller was held in the hand to move the hand avatar shown in Fig. 1c. For monitoring the muscle activity, seven Trigno™ wireless sEMG electrodes by Delsys® were attached to the muscles of the dominant upper arm. Each electrode produces five features.

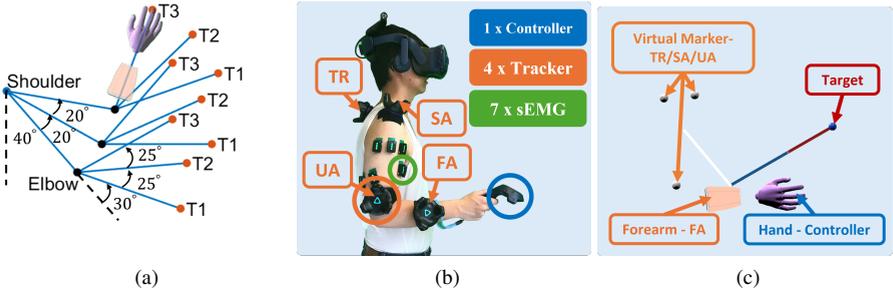


Figure 1: Offline dataset collection setup: (a) target shoulder and elbow poses; $T1$ – $T3$ denote three goals, (b) experimental setup and the placement of VIVE trackers and sEMG electrodes, and (c) VR avatar showing target example (side view).

For each goal, the subjects were tasked with 30 iterations of the forward-reaching. They were instructed to keep their final upper limb pose for one second upon reaching the goal. The data spanning from the initiation of the movement to the end of the holding period were reserved for feature extraction.

2.2.2 Human-In-The-Loop Experiments

In the HITL experiments, a real-world task is assessed in the VR environment, where subjects are asked to pick up and relocate clothespins by controlling a virtual powered prosthesis in real-time. The prosthesis is attached to their dominant side, as depicted in Fig. 2a. Such a task involves three prosthetic DoFs: elbow flexion/extension, wrist pronation/supination, and hand open/pinch. It is worth noting that this task involves different target poses (goals) compared to the offline experiment. Therefore, a different training dataset is first collected from the subject executing the clothespin relocation task to construct the GR system, followed by the HITL experiment. The training dataset involves the same features as in the offline experiment and consists of 10 iterations of data collection phases when the subject performs the clothespin relocation task using the avatar as displayed in Fig. 1c.

For real-time GR, features are extracted at a rate of 10 Hz and streamed as inputs to the GR system for controlling the prosthetic elbow and wrist movement. The hand open/pinch function typically requires a dedicated control algorithm or GR system separate from joint control due to the temporal sequence of gross arm movement and hand manipulation [2]. Thus, a switching mechanism is needed to transition between the two control algorithms, e.g., switching to hand control when detecting the upper limb joints are at rest [2]. In this work, to isolate the effects of switching between control algorithms, the hand open/pinch function is controlled through a button held in the non-dominant hand, as described in [19]. The socket of the prosthesis tracks the movement of the UA tracker and connects the prosthesis to the subject’s residual limb. The sensor setup mirrors the one used in the offline evaluation. Twelve more kinematic features are investigated, comprising the acceleration of the six DoF movements used in the

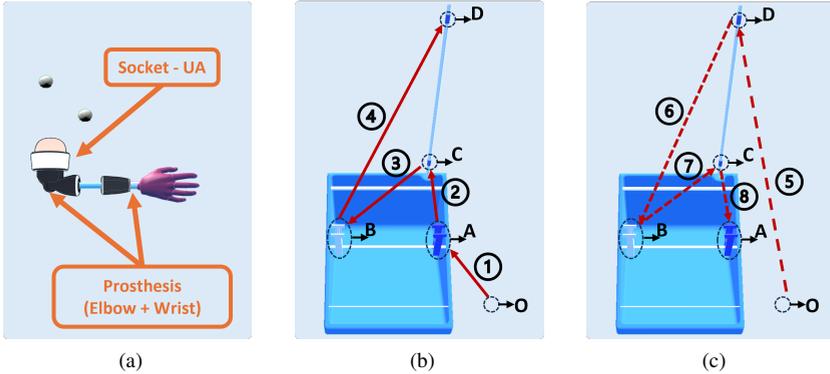


Figure 2: Human-in-the-loop experiment setup: (a) virtual 3-DoF prosthesis avatar, (b) forward stage of the RCRT task, and (c) backward stage of the RCRT task; numbers indicate goals (intended movements) with red solid and dashed arrows demonstrating the movement paths, letters A to D show the relocation positions, and “O” represents the arm resting position (upper-arm pointing downwards).

offline experiments and the kinematics of two additional DoFs, which are essential to extend the workspace from a plane (Fig. 1a) to a 3-dimensional space.

The HITL experiments adhere to the widely used Refined Clothespin Relocation Task (RCRT) documented in [14, 19], with a dedicated one-to-one scale virtual setup. The task comprises eight kinds of movements, representing eight distinct goals, accomplished in two stages. In the forward stage, see Fig. 2b, the subjects begin from the upper-arm resting and pointing downwards (denoted as “O” in Fig. 2b). They then sequentially pick up the two clothespins placed vertically on the horizontal rod (positions A and B) and transfer them to the vertical rod (positions C and D). Subsequently, in the backward stage, they start from the resting position and then return the clothespins at positions C and D to the original positions A and B, see Fig. 2c. The desired movements and goal categories are marked using red arrow lines and corresponding numbers; solid lines are used to depict desired movement trajectories in the forward stage, while dashed lines capture the desired movement trajectories for the backward stage of the experiment. From a GR perspective, which typically focuses on distinguishing different goals from the same initial state, the eight movement trajectories present three GR challenges: identifying whether the subject aims for A or D from initial position O; identifying whether the arm is moving toward C or D from position B; and identifying whether the movement is toward A or B from position C.

3 Related Work

Existing approaches in GR can be classified into three principal categories: the plan library-based GR approaches [16], the planning-based GR approaches [7, 27], and the data-driven GR approaches [22, 26, 30]. A plan library-based GR approach relies on

pre-defined libraries of plans, usually crafted by domain experts, designed to encode how candidate goals are meant to be achieved. Such a method works by comparing the observed agent’s actions with the plans from the libraries. A planning-based GR approach uses automated planning techniques, or planners, to generate optimal plans for achieving candidate goals.⁷ These planners usually rely on well-defined domain models that describe the environment in which the agent operates, the actions the agent can perform, and the effects of these actions. Once optimal plans for achieving the candidate goals are generated, the method compares them with the action sequence performed by the agent to assess how closely the agent’s actions align with the generated plans. If the observed actions closely match an optimal plan, the goal achieved by executing that plan is considered as a likely goal the agent is aiming for. Finally, data-driven approaches utilize historical data of agents’ actions to *learn* models that describe the principles for achieving the goals. When deducing likely goals, they leverage patterns and trends identified within the learned models to guide their inferences.

In the field of powered transhumeral prosthetics, existing literature has demonstrated that using varied features customized for individual subjects enhances the accuracy of identifying intended movements [37]. Such customization introduces complexity into the development of ideal plan libraries or domain models, necessitating unique plans or models for each subject. Data-driven GR techniques are well-suited for these customizations, as they can learn *personalized* patterns from the historical behavior of a patient to make more accurate goal inferences tailored to the individual.

Min et al. [22] proposed a GR approach based on LSTM neural networks. Once a network is trained on historical data, the approach identifies the most likely goal of the agent based on a newly observed sequence of actions performed by the agent. The PM-based GR system uses process discovery techniques to learn process models that describe the skills for accomplishing the candidate goals from the historical sequences of the agent’s actions. Subsequently, it relies on conformance checking techniques to examine commonalities and discrepancies between a newly observed sequence of actions and each learned model. These commonalities and discrepancies are then translated into a probability distribution over the candidate goals, capturing the likelihoods that the actions aim to reach these goals [26, 30]. Both the LSTM network-based GR and the PM-based GR techniques are data-driven, as they rely on the analysis of data generated by actions performed by the agent.

Machine learning techniques have been used to implement accurate prosthesis control [29]. Due to their robustness, machine learning classifiers are commonly implemented in upper-limb prostheses [9]. Given an input signal from sensors, a classifier predicts the output signal and the intended movement of the patient. Linear discriminant analysis (LDA) is probably the most commonly used classifier algorithm in prosthesis control, also used in gesture recognition scenarios via Myoelectric interfaces [15, 20]. Due to the lightweight and low-complexity nature of the LDA classifier, it can achieve high control accuracy with short training and processing times [25]. In our recent work, we confirmed that LDA can discriminate target poses reliably [37]. Recent works explore the use of artificial neural networks in prosthesis control. Neural

⁷The term “candidate goals” refers to a collection of possible goals from which GR techniques need to identify the most likely goal(s) that the agent aims to achieve.

networks achieve high recognition accuracy as they can learn complex dependencies between signal input and control outputs but require extensive training [29]. For instance, Huang et al. [13] successfully used LSTM neural networks to predict target poses based on time series of electromyography signals.

The existing approaches for detecting target poses based on LDA classifiers ignore the movement trajectory and rely only on signals registered at the target poses, which may lead to inferior recognition accuracy [37]. Meanwhile, the LSTM network-based approaches typically require large amounts of training data for reasonable performance [22]. Considering the need for individual customizations of GR inferences in the domain of transhumeral prostheses [37], it may be challenging to obtain sufficient volumes of training data from an individual subject. The PM-based GR approach, on the contrary, leverages data acquired along entire movement trajectories for training and can be less data-demanding [32].

This research aims to explore how the PM-based GR system can contribute to the development of powered transhumeral prostheses. Our previous work [32] demonstrated that combining clustering and PM-based GR techniques could address challenges in prosthesis development. This study improves the method by incorporating a pre-trained classifier, enabling the technique to be applied in real-time and interact with patients to complete tasks. We use LDA [37] and LSTM [13, 22] techniques as performance baselines. The evaluation is conducted both offline and online. In the offline setting, the evaluated GR techniques are compared using the pre-recorded data. In the online setting, we evaluate the performance of the GR techniques using human-in-the-loop control assessment. Hence, we address the concern raised in the literature considering the conflicting results regarding the performance correlation of the two conditions [12, 21, 24].

4 Approach

This section introduces two GR approaches based on process mining techniques for addressing the target pose recognition problem in the transhumeral prosthesis scenario (Section 4.1). One PM-based GR approach integrates hierarchical clustering and K -means clustering algorithms for feature selection and event discretization. The other approach involves segmenting the trajectory of data measurements and training an LDA classifier to distinguish data points for each segment, thereby discretizing the corresponding data points. Additionally, this section introduces a new Linear Discriminant Analysis (LDA) classifier for mapping sensor data to target poses (Section 4.2).

4.1 Goal Recognition Using Process Mining

The PM-based GR framework was proposed in our previous works [26, 30]. In this article, we extend it to allow goal inference based on sensor data, such as data generated by sEMG and kinematic movement sensors used to inform powered transhumeral prostheses. To apply the PM-based GR technique to the transhumeral prostheses scenario, we introduce a step to convert continuous, real-valued sensor data into discrete

events. These converted event sequences are then used as input for the training and inference phases of the PM-based GR. These event sequences are commonly referred to as *traces* in process mining, and traces that lead to the same goal are typically collected in an *event log*. A PM-based GR system learns knowledge models from event logs that describe ways for accomplishing candidate goals. Specifically, a knowledge model is a process model that captures sequences of actions that, when executed, should lead to the accomplishment of a specific goal. Traces from an event log for a certain goal are used as input to a *process discovery* algorithm to construct a process model that represents but also generalizes the possible sequences of actions to achieve the corresponding goal, thereby describing the “skills” for achieving the goal. In this work, we use Directly Follows Miner [18] as a process discovery algorithm and represent the discovered models as Petri nets.

To recognize goals, a PM-based GR system performs *conformance checking*, which involves a set of techniques designed to efficiently compare and analyze differences between a trace and a process model. When a trace is provided as input to the GR system for inference, it is compared to the historical behavior captured in each discovered process model. Specifically, optimal alignments are constructed between the trace and each model. An *optimal alignment* reflects a closest match between the trace and the model, describing the commonalities and discrepancies between the two. An alignment can be represented as a table with two rows, where the top row specifies the trace and the bottom row captures a closest matching trace described by the model. Such a table provides a systematic way of comparing the actions in the two traces. Each column of the table defines a move in the alignment, with the presence or absence of a skip symbol “ \gg ” indicating whether the move is asynchronous or synchronous, respectively.

Table 1: An example alignment.

	move 1	move 2	move 3
observation	action A	\gg	action B
model trace	action A	action C	\gg

Table 1 shows an example alignment between a sequence of observed actions ⟨action A, action B⟩ (refer to the top row in the table) and model trace ⟨action A, action C⟩ (bottom row). The example alignment has three moves (three columns). In a synchronous move, both sequences progress by executing the same action. In move 1 (first column), for instance, both sequences execute action A. In an asynchronous move, an action from one sequence is not matched by an action from the other sequence, which is denoted by the skip symbol. If the skip symbol appears in the observation row, such as in move 2 in Table 1 (second column), then the corresponding action in the model trace is not matched by observed actions; in the example, action C in the model trace is not matched by an action in the observation row. If the skip symbol appears in the model trace row, such as in move 3 in the example alignment (third column), then the corresponding action in the observed sequence of actions has no matching action in the model trace; in the example, observed action B has no matching action in the model trace. If one assigns a positive cost to every asynchronous move and no cost to all synchronous moves, then an *optimal alignment* between a sequence of observed actions

and a process model is an alignment between this observed sequence and some trace described by the model that yields the minimal cost, where the *cost* of an alignment is the sum of costs of all its moves.

Given optimal alignments between the input sequence of actions and each discovered process model (one model per candidate goal), the PM-based GR system computes the *weight* of each alignment by analyzing the types and patterns of moves within the alignment. While alignment cost serves as the basis for discovering alignments with minimal structural discrepancies between the compared traces, alignment weight quantitatively indicates the extent to which the observed trace aims to achieve the corresponding goal [30]. This alignment weight between trace τ and process model M_G for goal G is defined as follows:

$$\omega(\tau, M_G) = \phi + \lambda^m \times \sum_{i=1}^n (i^\delta \times c(\tau, M_G, i)). \quad (1)$$

In Eq. (1), the term $c(\tau, M_G, i)$ represents the cost of the move in the optimal alignment between τ and M_G at position i . To handle partially observed input sequences of actions toward goals, when computing alignment weights, we assign a cost of one to all asynchronous moves with the skip symbol in the model trace and a cost of zero to all other moves. The length of the optimal alignment is denoted by n . The constant term ϕ is a smoothing factor, aiming to mitigate overconfidence in inferring the goal during the initial stages of movement, especially when the sensors have only collected a limited number of data points. The discount factor δ amplifies the importance of later asynchronous moves by giving more weight to recent moves in the alignment. Moreover, the parameter $\lambda \geq 1$ functions as a penalty for consecutive asynchronous move suffixes in the alignment. This penalty is applied when there are m consecutive asynchronous moves that are not matched by the actions in the model trace at the end of the alignment. In this work, when computing alignment weights, we use default parameters of $\phi = 50$, $\lambda = 1.5$, and $\delta = 1$. To ensure consistent analysis, the same parameter settings must be used when checking the conformance between an input trace and all the process models. For more details on the computation of alignment weights, see our previous work [30].

Given the weights of alignments between the input observed sequence of actions and all the discovered process models, the PM-based GR system computes the probability distribution over the candidate goals. The probability of each goal indicates the likelihood that the input sequence of actions aims to achieve that goal. Based on this distribution, the PM-based GR system then constructs and returns the set of most likely goals as the inference result. Specifically, the probability of achieving candidate goal G based on the input sequence of actions τ is computed as follows [30]:

$$\Pr(G \mid \tau) = \frac{e^{-\beta \times \omega(\tau, M_G)}}{\sum_{G' \in \mathcal{G}} e^{-\beta \times \omega(\tau, M_{G'})}}. \quad (2)$$

In Eq. (2), \mathcal{G} is the set of all candidate goals and β is the level of trust in the “learned”

process models, which is defined as follows:

$$\beta = \frac{1}{1 + \min_{G \in \mathcal{G}} \omega(\tau, M_G)}. \quad (3)$$

In this work, we aim to recognize the intended pose of a subject based on multi-dimensional, real-valued features collected by sensors. We use two data conversion approaches to transform the sensor data into traces of events. The first approach that uses clustering, for instance, hierarchical clustering or K -means clustering, was presented in our previous work Section 4.1.1. The second approach, involving training an LDA classifier leveraging the target pose information in the training data, is new and is presented in Section 4.1.2.

4.1.1 Process Mining-Based Goal Recognition Using Clustering

In this section, we use a running example to illustrate how the PM-based GR framework combines feature selection and event discretization techniques to tackle the target pose prediction problem. In the example, we instructed a subject to perform six iterations of reaching tasks, three times to reach target $T1$ and three times to reach target $T2$. The GR system observed six sequences of signals, each comprising 30 continuous real-valued features (including sEMG signals and kinematic signals), denoted as f_1 to f_{30} . Traces 1 to 3 represent signal sequences recorded during movements toward target pose $T1$, while traces 4 to 6 represent sequences for movements toward $T2$. The dataset of input signal sequences and tools for reproducing the results are publicly available.⁸ Table 2 presents an extract of the example dataset, with each row containing collected feature values ordered by their respective timestamps of data collection. In the table, each row characterizes an action from the trace with the identifier specified in the *Trace* column that aims to accomplish the goal specified in the *Goal* column.

Table 2: Extract of the running example dataset.

Trace	Goal	f_1	f_2	f_3	...	f_{29}	f_{30}
1	T1	5.19727337	7.02395793	0.00254431	...	5.39759498	-0.3722619
1	T1	7.76278776	8.08816201	0.00472689	...	1.01557531	1.37592798
1	T1	13.4185557	8.87159453	0.00821896	...	-4.0004147	1.65328609
1	T1	22.0916619	9.04377674	0.01015369	...	-5.5399488	-1.7805512
1	T1	31.3641039	9.3586209	0.009165	...	-3.5156837	1.36367015
1	T1	38.2312577	10.139119	0.00616715	...	-1.4720033	5.87820456
1	T1	42.0592085	10.8827908	0.00315491	...	-0.3338844	4.29640897
2	T1	7.39110795	6.07336937	0.00064332	...	2.92403705	1.46698529
2	T1	10.5229866	7.44734189	0.00194998	...	1.60034347	2.94734496
2	T1	17.6705947	8.62902577	0.00393832	...	1.41373702	2.84419105
...
6	T2	53.1712171	19.394227	0.00270796	...	-3.5619104	0.78601719
6	T2	60.1200614	22.6060167	0.00091891	...	-2.6239834	1.7355157
6	T2	64.1830578	25.2975943	-0.0003433	...	-1.1970367	0.92412363
6	T2	66.8916142	27.5304609	-0.0017204	...	0.31022101	0.95595258

The PM-based GR approach that uses clustering for event discretization comprises five steps. Next, we present these steps.

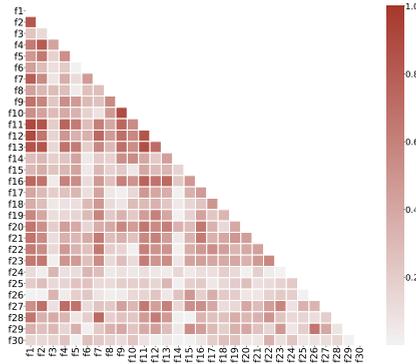


Figure 3: Correlation matrix.

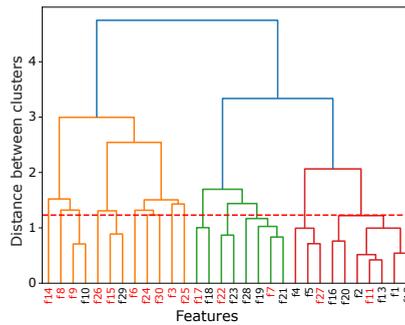


Figure 4: Dendrogram and selection of clusters. The red dotted line represents the threshold used to cut the dendrogram to form 15 clusters.

Step 1: The first step involves reducing the dimensionality of the input data by selecting features that have a substantial predictive power. Specifically, we exclude highly correlated features. First, we compute correlations between each pair of features in the dataset. Figure 3 summarizes the absolute values of the Pearson correlation coefficients for all pairs of features. Then, we use agglomerative hierarchical clustering [6] to group features into N_f clusters using correlation coefficients to define distances between the features. The resulting dendrogram, see Fig. 4, provides a visual representation of the hierarchical structure of the computed clusters based on the correlations from Fig. 3. One can use a threshold value to determine the number of clusters they want to extract from the dendrogram. The similarity threshold specifies the desired distance between formed clusters. As shown in Fig. 4, in our example, setting the threshold to 1.23 (see the red dashed horizontal line in the figure) allows us to extract 15 clusters; groups of features connected below the threshold are considered as one cluster, while groups of features above the threshold are separate clusters. From each extracted cluster of features, we then select one representative feature with the largest correlation with all

⁸<https://doi.org/10.26188/25487290>

the other features in the cluster. The selected features constitute a lower-dimensional representation of the dataset. In the running example, we reduced the feature space to $N_f = 15$ dimensions; the selected 15 features are highlighted in red in Fig. 4.

An extract of the reduced dataset is displayed in Table 3.

Table 3: Extract of the reduced running example dataset.

Trace	Goal	f_3	f_6	...	f_{27}	f_{30}	Event
1	T1	0.002544311	0.121301538	...	9.057075924	-0.372261852	e_0
1	T1	0.004726894	0.210557727	...	10.55266625	1.375927982	e_6
1	T1	0.008218956	0.391712037	...	4.416704571	1.653286092	e_5
1	T1	0.010153689	0.327711654	...	1.372325318	-1.780551234	e_2
1	T1	0.009165	0.311548058	...	5.526959903	1.363670147	e_5
1	T1	0.00616715	0.734098175	...	8.869853729	5.87820456	e_6
1	T1	0.003154906	1.227944362	...	4.916643199	4.296408971	e_8
2	T1	0.000643317	0.103164637	...	12.86040763	1.466985286	e_8
2	T1	0.001949978	0.336630569	...	13.45204634	2.94734496	e_6
2	T1	2.00E-05	2.91E-05	...	0.65070719	88.8458582	e_2
...
6	T2	3.47E-05	1.25E-05	...	0.16984547	90.8573749	e_1
6	T2	3.85E-05	1.06E-05	...	0.30483842	50.9656092	e_1
6	T2	-0.000343322	0.410761081	...	23.78469914	0.924123631	e_4
6	T2	-0.001720367	0.504684583	...	21.08908217	0.955952575	e_3

Step 2: Next, we convert the reduced dataset into event traces. An *event trace* is a sequence of discrete events, each identified by a label (shown in the last column of Table 3), referring to the same trace identifier and ordered by their timestamps. In contrast, the corresponding original trace consists of feature values at each timestamp, represented as a set of real numbers. To convert a trace into an event trace, we perform K -means clustering [11] over N_f -dimensional data points to obtain N_c clusters. The K -means algorithm groups similar data points together to minimize the variance within each cluster and maximize the distance between different clusters. Within each cluster, the data points are considered as instances of the same event.

The K -means algorithm takes the number of clusters it constructs as input. The approach we use to determine the appropriate number of clusters is elaborated in Section 5. For the purpose of demonstration, in the running example, we use $N_c = 10$. Consequently, the 15-dimensional data points are grouped into 10 clusters, represented by events e_0 to e_9 , as shown in the “Event” column in Table 3. Then, we split the obtained traces of events into event logs, where each event log contains all the traces toward a specific candidate goal. As there are two target poses in the running example, we split the traces into two event logs: L_1 and L_2 . The traces in the event logs L_1 and L_2 aim towards target poses $T1$ and $T2$, respectively.

Step 3: In this step, we use process discovery techniques to construct process models from the event logs obtained in the previous step. Figures 5 and 6 depict Petri nets M_1 and M_2 constructed using Directly Follows Miner [18] from event logs L_1 and L_2 , respectively. Models M_1 and M_2 , hence, describe the processes for reaching target poses $T1$ and $T2$, respectively. They constitute “knowledge” learned from historical experiences stored in our GR system.

Step 4: Next, we use conformance checking techniques [33] to assess the commonali-

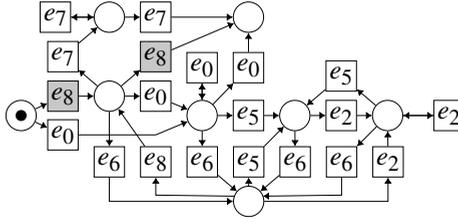


Figure 5: Process model M_1 discovered from even log L_1 .

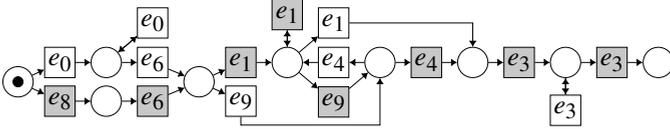


Figure 6: Process model M_2 discovered from even log L_2 .

ties and discrepancies between the discovered models and observed traces, leveraging this information to infer the intended target poses. It is important to note that clustering approaches, like K -means, do not inherently generate classification models for original traces of sensor measurements. In the running example, we choose a sequence of signals from Table 4 as the testing trace. Each signal is a multi-dimensional data point comprising real-valued measurements of various sEMG and kinematic features. The testing trace is partial, containing only the first six measurements representing movement toward a target pose. This partial trace simulates the condition when the patient has commenced the movement but has not yet reached the target pose.

Table 4: An example partial sequence of signals observed by the GR system.

Timestamps	f_1	f_2	f_3	...	f_{29}	f_{30}
1	6.003909937	7.491469679	-0.000426667	...	3.003725052	1.044722093
2	10.4000259	8.661748533	0.002236012	...	-1.678683223	1.119218147
3	18.34086353	9.394745911	0.005143733	...	-4.419588229	0.414732289
4	31.05712269	10.02530957	0.006663839	...	-5.512023759	-0.470670561
5	45.26542356	11.12347556	0.005400478	...	-5.503936393	-0.967709384
6	56.95525497	13.56039176	0.002532783	...	-4.995283183	1.20365095

Given all the historical traces used for training process models, as shown in Table 2, and the testing trace, as shown in Table 4, the GR system extracts features following the approach outlined in step 1 and converts the multi-dimensional data points into discrete events by applying K -means clustering to all data points from both the training and testing traces, as described in step 2. The obtained discrete events are ordered according to their timestamps into event traces. In our running example, the testing trace is converted into trace τ as follows:

$$\tau = \langle e_8, e_6, e_2, e_1, e_1, e_9 \rangle.$$

Given trace τ and process models M_1 and M_2 discovered in the previous step, the GR system computes optimal alignments σ_1 and σ_2 between the trace and the models.

The constructed optimal alignments are shown below. The transitions of M_1 and M_2 involved in the alignments are shaded in gray in Figs. 5 and 6.

$$\sigma_1 = \frac{\begin{array}{|c|c|c|c|c|c|c|c|} \hline \tau & e_8 & \gg & e_6 & e_2 & e_1 & e_1 & e_9 \\ \hline \end{array}}{\begin{array}{|c|c|c|c|c|c|c|c|} \hline M_1 & e_8 & e_8 & \gg & \gg & \gg & \gg & \gg \\ \hline \end{array}}$$

$$\sigma_2 = \frac{\begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline \tau & e_8 & e_6 & e_2 & e_1 & e_1 & e_9 & \gg & \gg & \gg \\ \hline \end{array}}{\begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline M_2 & e_8 & e_6 & \gg & e_1 & e_1 & e_9 & e_4 & e_3 & e_3 \\ \hline \end{array}}$$

Step 5: Finally, in the last step, the PM-based GR system leverages information on moves in the constructed optimal alignments to calculate the probability distribution over the candidate target poses. The probabilities associated with each target pose indicate the likelihood of the subject aiming to reach that pose. To compute the probability distribution, the system first computes alignment weights between trace τ and models M_1 and M_2 according to Eq. (1). Then, it uses Eqs. (2) and (3) to compute the probability of reaching every target pose. In our running example, given the two alignments σ_1 and σ_2 , the probabilities of the subject that induced trace τ aiming to reach target poses $T1$ and $T2$ are 0.06 and 0.94, respectively. Consequently, the PM-based GR system infers that the subject aims to reach $T2$.

4.1.2 Process Mining-Based Goal Recognition Using LDA Classifier

This section describes an alternative labeling technique utilizing a trained LDA classifier to assign labels to signals. An LDA classifier can map multi-dimensional values to particular categories. We assume that the data points in the trajectories can be divided into sub-groups based on their timestamps. For example, data points collected early in the trajectory may differ significantly from those collected later. Therefore, we aim to divide the data points into sub-groups according to their time order and then train an LDA classifier to identify which sub-group a data point belongs to. This approach allows the classifier to distinguish whether a data point was collected in the early or late stage of the trajectory. Once trained, the classifier is applied to convert multi-dimensional, real-valued signals into discrete labels (or events). The subsequent steps are the same as steps 3–5 outlined in Section 4.1.1. Specifically, we proceed by creating event logs of event traces toward various goals and use a process discovery technique to construct process models. With these process models, we conduct conformance checking to construct optimal alignments between the process models and a newly observed trace of actions. Finally, we utilize these optimal alignments to compute a distribution over candidate goals and to infer the most likely intended poses.

We use another running example to illustrate the PM-based GR approach based on an LDA classifier. In this example, we start with 20 signal sequences recorded toward target poses $T1$ and $T2$, with 10 signal sequences per target pose.⁹

Next, we present the four steps of the PM-based GR approach that uses an LDA classifier for event discretization.

⁹The code and data for replicating this running example can be accessed here: <https://doi.org/10.26188/25487290>.

Step 1: In this step, signals of measurements are converted into discrete events, which is achieved by training an LDA classifier. During the training phase, we label the data points according to two aspects: the part of the trajectory the data points relate to and the final target pose reached by the corresponding movement. For instance, consider a sequence of signals illustrated in Fig. 7, which eventually reaches target pose $T2$. During the movement, 10 data points are collected. We divide the sequence into five segments based on the timestamps of the collected data points. Note that the number of segments to use is a parameter, and for illustration purposes, we use five segments in our example. We then assign each data point a label of the format T_xP_y , where T_x is the target pose that is reached by the sequence of signals the data point belongs to and P_y is part of the trace the data point is located at. For example, the first part of the sequence of signals from Fig. 7 capturing the initial phase of the movement contains two data points. Consequently, these data points are assigned label $T2P1$.

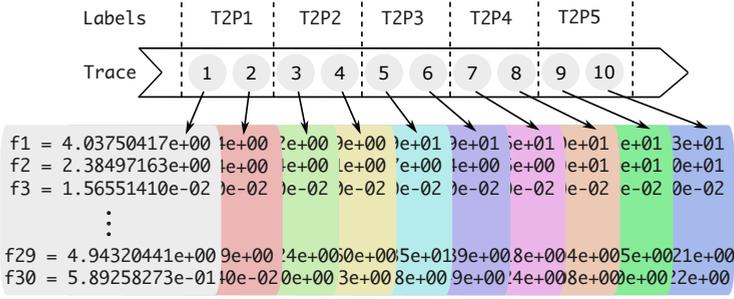


Figure 7: Labeling a sequence of signals.

Once all the data points are labeled, we use the obtained labels to train an LDA classifier. Once we obtain this classifier, we use it to convert the original multi-dimensional signals into discrete labels representing events. We then group the obtained events into event traces, where each trace captures all events from one movement toward a target pose. The traces are then grouped into event logs, where each event log contains all the constructed traces to a particular goal. In the running example, 20 sequences of signals are converted into two event logs L'_1 and L'_2 , as shown below:

$$\begin{aligned}
 L'_1 = \{ & \langle T2P1, T1P1, T1P2, T1P2, T1P3, T1P4, T1P5, T1P5 \rangle, \\
 & \langle T2P1, T1P1, T1P1, T2P1, T1P2, T1P2, T1P3, T1P3, T1P4, T1P4, T1P5, T1P5 \rangle, \\
 & \langle T1P1, T1P2, T1P2, T1P2, T1P3, T1P3, T1P4, T1P4, T1P5 \rangle, \\
 & \langle T2P1, T1P2, T1P2, T1P3, T1P4, T1P4, T1P5 \rangle, \\
 & \langle T1P1, T2P1, T2P2, T1P2, T2P3, T1P3, T1P4, T1P4, T1P5, T1P5 \rangle, \\
 & \langle T1P1, T1P1, T1P2, T1P2, T1P3, T1P3, T1P4, T1P5 \rangle, \\
 & \langle T1P1, T1P1, T1P2, T1P2, T1P3, T1P3, T1P4, T1P4, T1P5 \rangle, \\
 & \langle T1P1, T1P1, T1P1, T1P2, T1P2, T1P3, T1P3, T1P3, T1P4, T1P5, T1P5 \rangle, \\
 & \langle T1P1, T1P1, T1P2, T1P2, T1P3, T1P3, T1P4, T1P4, T1P5 \rangle, \\
 & \langle T1P1, T1P1, T1P2, T1P2, T1P3, T1P3, T1P4, T1P4 \rangle \},
 \end{aligned}$$

$$L'_2 = \{ \langle T2P1, T2P1, T2P2, T2P2, T2P3, T2P3, T2P4, T2P4, T2P5, T2P5 \rangle, \\ \langle T2P1, T2P1, T2P2, T2P2, T2P3, T2P3, T2P4, T2P4, T2P5 \rangle, \\ \langle T2P1, T2P1, T2P1, T2P1, T2P1, T2P2, T2P2, T2P3, T2P3, T2P4, T2P4, T2P5 \rangle, \\ \langle T2P1, T2P1, T1P1, T1P2, T2P2, T2P3, T2P3, T2P4, T2P4, T2P5, T2P5 \rangle, \\ \langle T2P1, T2P1, T2P1, T2P2, T2P3, T2P3, T2P4, T2P4, T2P4 \rangle, \\ \langle T2P1, T2P1, T2P2, T2P2, T2P3, T2P3, T2P4, T2P4, T2P5, T2P5 \rangle, \\ \langle T2P1, T1P1, T2P2, T2P2, T2P3, T2P3, T2P4, T2P5, T2P5, T2P5 \rangle, \\ \langle T2P1, T2P2, T2P2, T2P3, T2P4, T2P4, T2P5, T2P5 \rangle, \\ \langle T2P1, T2P1, T1P2, T2P2, T2P3, T2P3, T1P4, T2P4, T2P5, T2P5 \rangle, \\ \langle T1P1, T1P1, T1P2, T1P2, T2P3, T2P3, T2P3, T2P4, T2P4, T2P5 \rangle \}.$$

Step 2: In this step, we use Directly Follows Miner to discover process models from all event log obtained in step 1. For instance, using event logs L'_1 and L'_2 , we construct process models M'_1 and M'_2 shown in Figs. 8 and 9, respectively, representing the “knowledge” about how target poses $T1$ and $T2$ can be reached.

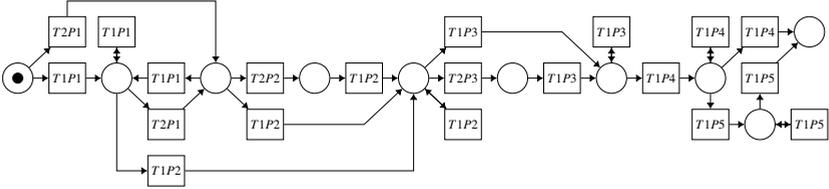


Figure 8: Process model M'_1 discovered from event log L'_1 .

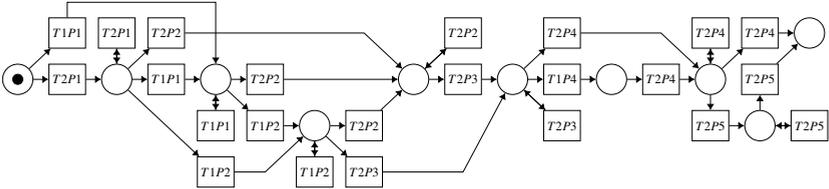


Figure 9: Process model M'_2 discovered from event log L'_2 .

Step 3: In this step, we conduct conformance checking to identify discrepancies between a newly observed trace and the discovered models. Once the GR system receives a new sequence of sensor data points recorded during a subject’s movement, we use the trained classifier from step 1 to map the data points to events, resulting in a trace of events. For illustration purposes, consider we obtain event trace τ' shown below.

$$\tau' = \langle T1P1, T1P1, T2P1, T2P3, T2P3, T2P3, T2P4 \rangle.$$

Note that τ' is a prefix of an entire trace representing the initial phase of the movement toward a target pose. The optimal alignments between τ' and models M'_1 and M'_2 are sequences of moves σ'_1 and σ'_2 shown below.

Step 4: Finally, in the last step, we compute the probabilities that the subject aims to reach target poses $T1$ and $T2$ using the diagnoses of the synchronous and asynchronous

$$\sigma'_1 = \frac{\tau' \mid T1P1 \mid T1P1 \mid T2P1 \mid \gg \mid T2P3 \mid \gg \mid \gg \mid T2P3 \mid T2P3 \mid T2P4}{M'_1 \mid T1P1 \mid T1P1 \mid T2P1 \mid T1P2 \mid T2P3 \mid T1P3 \mid T1P4 \mid \gg \mid \gg \mid \gg}$$

$$\sigma'_2 = \frac{\tau' \mid T1P1 \mid T1P1 \mid \gg \mid T2P1 \mid T2P3 \mid T2P3 \mid T2P3 \mid T2P4}{M'_2 \mid T1P1 \mid T1P1 \mid T2P2 \mid \gg \mid T2P3 \mid T2P3 \mid T2P3 \mid T2P4}$$

moves in the optimal alignments σ'_1 and σ'_2 . Assuming the asynchronous moves with skips in the model have the cost of one, and using the default parameters for the GR system, the alignment weights computed following Eq. (1) are 110.75 and 53, respectively. Then, we use Eq. (2) and Eq. (3) to compute the probabilities of the subject reaching $T1$ and $T2$ after observing trace τ' , which are equal to 0.26 and 0.74, respectively. These probabilities indicate that, based on the sensor data, the subject is more likely to reach target pose $T2$.

4.2 Goal Recognition Using Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is used as a baseline approach for prosthetic pose recognition. This LDA baseline leaves room for improvement. One such improvement we discuss here. A state-of-the-art method for prosthetic pose recognition trains an LDA classifier to map input signals to specific event labels [38]. In that approach, the LDA classifier is trained using data collected while subjects hold their arms in target poses. As a result, this method may perform poorly when predicting target poses from signals captured during arm movements toward the target. To improve pose prediction under these conditions, we modify the LDA training phase to incorporate signals collected *during* arm movements.

Suppose sensors capture a sequence of 15 data points while a subject moves their arm to reach a pose and stops at that pose; see Fig. 10. Each data point is a collection of features. This sequence of data points can be divided into two phases. For instance, data points 1 through 10 were captured while the arm was moving toward the target pose, while the last five data points, data points 11 through 15, were collected when the arm was fixed after reaching the target position. In the work by Yu et al. [38], the LDA classifier is trained with the last five data points. We adjust their approach by training the LDA classifier using data points 1 to 10. We refer to this method as *dynamic LDA* since it is trained on the data obtained during the movement of the subject, reflecting the signal patterns during the movement. Consequently, we refer to the original approach as *static LDA*, as it is trained on the data collected while the subject is stationary in the target pose.

Note that there are no customizable parameters in this approach. It is trained using all data points along the trajectory, and once the subject reaches a target, any remaining data points are discarded. Both methods, static LDA and dynamic LDA, are used as baselines for comparison with the PM-based GR techniques.

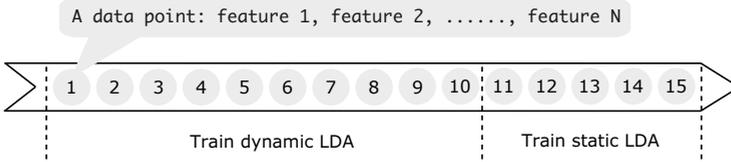


Figure 10: A sequence of data points captured by sensors.

5 Evaluation

In this section, we compare the performance of our PM-based GR approaches for recognizing target poses with three baselines. The experiments are conducted in two settings: offline experiments, as described in Section 2.2.1, and human-in-the-loop (HITL) experiments, as detailed in Section 2.2.2. In the offline experiments, we evaluate two PM-based GR approaches presented in Section 4.1 by comparing their performance with three baselines, namely the LSTM-based approach [13], the static LDA classifier [38], and the dynamic LDA classifier (an improved version of the static LDA approach described in Section 4.2). In the HITL experiments, we compare the two best-performing approaches from the offline experiments: the PM-based approach that uses an LDA classifier to convert sensor data into events and the dynamic LDA classifier. When compared, the different GR approaches were trained using (possibly different parts of) data collected during the same arm movements toward target poses.

For convenience, we denote the five evaluated GR approaches as follows:

1. $PM_{clustering}$ denotes the PM-based GR approach that uses clustering to convert sensor data into events;
2. $PM_{classifier}$ indicates the PM-based GR approach that uses an LDA classifier to convert sensor data into events;
3. $LSTM$ signifies the LSTM-based approach for target pose recognition;
4. $sLDA$ represents the static LDA classifier for target pose recognition; and
5. $dLDA$ denotes the dynamic LDA classifier for target pose recognition.

5.1 Performance Measures

To assess the quality of goal inferences by the evaluated techniques, we use the F_1 score and balanced accuracy. These measures are computed based on four terms: True Positive, True Negative, False Positive, and False Negative. The True Positive (TP) term denotes the number of correct goals inferred by the GR system. The True Negative (TN) component is the number of incorrect goals that were not inferred. The False Positive (FP) term represents the number of incorrect goals inferred by a GR system. Finally, the False Negative (FN) component refers to the number of correct goals that were not recognized by the system.

Given the four terms, the F_1 score is computed as follows [5]:

$$F_1 = \frac{2 \times TP}{2 \times TP + FP + FN}.$$

Balanced accuracy (*bacc*) is computed as follows [4]:

$$bacc = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right).$$

The F_1 score is also known as the harmonic mean of precision and recall, where precision is the fraction of the correctly inferred target poses among the total number of poses that were inferred, and recall is the fraction of the correctly inferred target poses among the relevant poses. Balanced accuracy is well suited for measuring performance in imbalanced scenarios, like the case when there is only one true target pose.

Both offline and HITL experiments include multiple GR problem instances, where a problem instance comprises one attempt to recognize the target pose at some stage during the movement toward that pose. For each problem instance, we compute F_1 score and balanced accuracy, subsequently calculating the averages across all problem instances for each individual subject.

5.2 Baselines

We compare our PM-based GR approaches with three baselines: the LSTM-based approach, the static LDA approach, and the dynamic LDA approach.

LSTM neural networks, tailored to recognize dependencies and patterns in sequential data, have proven exceptionally adept at classifying multi-dimensional, continuous, real-value measurements, such as sEMG and kinematic sensor data. In our experiments, we adopt configurations and hyperparameters outlined in [13] for implementing the LSTM-based GR baseline.

LDA functions are trainable classifiers that utilize linear decision boundaries to categorize multi-dimensional, continuous, real-valued data points into predefined clusters. They are effective in analyzing individual data points, such as sEMG and kinematic signals, captured at specific moments. For our experiments, we implemented the static LDA classifier baseline detailed in [37] and the dynamic LDA classifier baseline discussed in Section 4.2. The static and dynamic LDA classifiers differ in the data that is used for training. The static LDA classifier is trained on data collected from the arm being held in target poses, using ten data points from each pose. The dynamic LDA classifier, however, is trained on data collected during the movement of the arm toward the target poses, capturing the arm’s motion dynamics.

5.3 Offline Experiments

The offline experiments required collecting both training and testing data together. Ten subjects (subject IDs 1 to 10) participated in the experiments, completing three tasks. Each task involved moving their hands to reach one of three target positions: T_1 , T_2 , or T_3 , as described in Section 2.2.1. Each subject was asked to perform each reaching task 30 times, resulting in a total of 90 trajectories per subject. These 90 traces included both training and testing data. We evaluated recognition performance at an individual level using cross-validation. The 90 trajectories collected for each subject were split into training and testing sets, with 87 traces used for training three process models, one model per goal discovered using 29 traces, and the remaining trace used for testing the

Table 5: Average F_1 score and balanced accuracy ($bacc$) for individual subjects at different levels of observation (highest in bold).

Subject ID	Features	Clusters	Obs.	$PM_{clustering}$		$LSTM$		$sLDA$		$dLDA$		$PM_{classifier}$	
				F_1 score	$bacc$	F_1 score	$bacc$	F_1 score	$bacc$	F_1 score	$bacc$	F_1 score	$bacc$
1	29	70	10%	0.519	0.525	0.356	0.517	0.333	0.500	0.544	0.658	0.533	0.547
			30%	0.515	0.567	0.444	0.583	0.344	0.508	0.633	0.725	0.604	0.700
			50%	0.570	0.656	0.389	0.542	0.367	0.525	0.689	0.767	0.744	0.808
			70%	0.691	0.758	0.544	0.658	0.467	0.600	0.756	0.817	0.796	0.847
2	1	10	10%	0.489	0.497	0.311	0.483	0.333	0.500	0.311	0.483	0.333	0.486
			30%	0.504	0.533	0.333	0.500	0.333	0.500	0.444	0.583	0.422	0.561
			50%	0.544	0.558	0.333	0.500	0.378	0.533	0.511	0.633	0.519	0.631
			70%	0.572	0.589	0.333	0.500	0.422	0.567	0.522	0.642	0.530	0.647
3	2	150	10%	0.491	0.494	0.322	0.492	0.267	0.450	0.344	0.508	0.494	0.506
			30%	0.563	0.597	0.356	0.517	0.322	0.492	0.456	0.592	0.528	0.628
			50%	0.604	0.678	0.311	0.483	0.367	0.525	0.600	0.700	0.552	0.650
			70%	0.652	0.731	0.333	0.500	0.422	0.567	0.567	0.675	0.633	0.714
4	34	50	10%	0.498	0.508	0.444	0.583	0.389	0.542	0.511	0.633	0.467	0.492
			30%	0.500	0.533	0.467	0.600	0.344	0.508	0.556	0.667	0.589	0.681
			50%	0.519	0.572	0.489	0.617	0.400	0.550	0.500	0.625	0.541	0.639
			70%	0.589	0.653	0.511	0.633	0.567	0.675	0.578	0.683	0.622	0.708
5	32	90	10%	0.511	0.531	0.356	0.517	0.244	0.433	0.544	0.658	0.556	0.614
			30%	0.617	0.686	0.478	0.608	0.289	0.467	0.522	0.642	0.619	0.700
			50%	0.630	0.708	0.400	0.550	0.300	0.475	0.578	0.683	0.604	0.700
			70%	0.811	0.858	0.522	0.642	0.556	0.667	0.544	0.658	0.622	0.714
6	28	160	10%	0.483	0.497	0.367	0.525	0.367	0.525	0.400	0.550	0.478	0.528
			30%	0.496	0.567	0.378	0.533	0.344	0.508	0.456	0.592	0.511	0.617
			50%	0.496	0.592	0.422	0.567	0.378	0.533	0.567	0.675	0.574	0.675
			70%	0.600	0.681	0.489	0.617	0.622	0.717	0.589	0.692	0.633	0.725
7	22	80	10%	0.522	0.542	0.367	0.525	0.322	0.492	0.467	0.600	0.528	0.597
			30%	0.493	0.550	0.422	0.567	0.344	0.508	0.522	0.642	0.619	0.694
			50%	0.520	0.608	0.400	0.550	0.456	0.592	0.733	0.800	0.696	0.769
			70%	0.554	0.631	0.400	0.550	0.522	0.642	0.600	0.700	0.704	0.769
8	34	100	10%	0.456	0.489	0.433	0.575	0.300	0.475	0.433	0.575	0.507	0.586
			30%	0.494	0.597	0.411	0.558	0.311	0.483	0.644	0.733	0.630	0.717
			50%	0.572	0.667	0.511	0.633	0.378	0.533	0.633	0.725	0.719	0.786
			70%	0.707	0.781	0.522	0.642	0.589	0.692	0.756	0.817	0.719	0.789
9	23	100	10%	0.498	0.503	0.411	0.558	0.400	0.550	0.500	0.625	0.506	0.519
			30%	0.528	0.564	0.356	0.517	0.344	0.508	0.589	0.692	0.557	0.658
			50%	0.457	0.542	0.344	0.508	0.367	0.525	0.611	0.708	0.607	0.700
			70%	0.467	0.558	0.411	0.558	0.567	0.675	0.678	0.758	0.704	0.778
10	28	170	10%	0.500	0.500	0.467	0.600	0.322	0.492	0.489	0.617	0.467	0.481
			30%	0.648	0.714	0.511	0.633	0.378	0.533	0.589	0.692	0.657	0.733
			50%	0.733	0.794	0.578	0.683	0.311	0.483	0.700	0.775	0.733	0.792
			70%	0.867	0.900	0.656	0.742	0.533	0.650	0.678	0.758	0.711	0.783
Average				0.562	0.613	0.422	0.567	0.390	0.543	0.559	0.669	0.589	0.667
Standard deviation				0.030	0.033	0.026	0.019	0.030	0.023	0.033	0.025	0.032	0.031

GR performance. This procedure was repeated 30 times for each subject, each time leaving one trace toward each of the three goals for testing.

The key parameters, number of selected features and number of clusters, as mentioned in Section 4.1.1, are determined through brute-force search. In the experimental dataset, which comprises 47 distinct features, we set the selection range for the number of features (N_f) to any integer from 1 to 47 inclusive. For the number of discrete event clusters (N_c), we tested values in the set $\{10, 20, \dots, 200\}$. To identify the optimal combination of N_f and N_c , we systematically evaluated each pair to identify the one that results in the best F_1 score. We then evaluated and compared all the GR techniques, namely $PM_{clustering}$, $PM_{classifier}$, $LSTM$, $sLDA$, and $dLDA$, using the same set of selected features. Each approach was implemented and deployed on a cloud server with a single core of an Intel[®] Xeon Processor at 2.0GHz. As GR techniques aim to identify goals before full sequences of signals are observed, we evaluated the approaches using prefixes of 10%, 30%, 50%, and 70% of the total number of sensor measurements.

Table 5 displays the average F_1 score and balanced accuracy ($bacc$) for recognizing

Table 6: Results of the t -tests for comparing average F_1 score.

	$PM_{clustering}$	$LSTM$	$sLDA$	$dLDA$	$PM_{classifier}$
$PM_{clustering}$		<i>diff</i>	<i>diff</i>	<i>diff</i>	—
$LSTM$	3.767e-07		<i>diff</i>	<i>diff</i>	<i>diff</i>
$sLDA$	6.244e-19	5.593e-14		<i>diff</i>	<i>diff</i>
$dLDA$	2.600e-03	6.741e-04	1.093e-19		—
$PM_{classifier}$	1.022e-01	3.437e-05	1.018e-18	1.358e-01	

Table 7: Results of the t -tests for comparing average balanced accuracy ($bacc$).

	$PM_{clustering}$	$LSTM$	$sLDA$	$dLDA$	$PM_{classifier}$
$PM_{clustering}$		<i>diff</i>	<i>diff</i>	—	—
$LSTM$	1.458e-02		<i>diff</i>	<i>diff</i>	—
$sLDA$	2.442e-10	5.593e-14		<i>diff</i>	<i>diff</i>
$dLDA$	4.579e-01	6.741e-04	1.093e-19		—
$PM_{classifier}$	3.397e-01	1.433e-01	1.426e-09	6.183e-01	

target poses of each subject. The “Features” and “Clusters” columns show the number of selected features, N_f , and the number of discrete event clusters, N_c , for each subject. The last row shows the average across all subjects and all levels of observation.

We conducted t -tests to assess the statistical significance of differences in the average F_1 scores and $bacc$ between evaluated GR techniques. The null hypothesis of each conducted t -test is that there is no significant difference between the average F_1 scores ($bacc$) of compared techniques. These t -tests, comparing five approaches based on the two performance measures, yield pairwise p -values, which are listed in Table 6 and Table 7. The entries below the diagonal in a table (the lower triangular part) show the p -values for the corresponding compared techniques, while the entries above the diagonal indicate whether the two approaches are significantly different from each other. If the p -value is less than 0.05, we use entry “*diff*” to represent that the two approaches are significantly different (the null hypothesis rejected); otherwise, we use “—” to represent that they are not significantly different (cannot reject the null hypothesis).¹⁰

The $PM_{classifier}$ technique achieves the highest F_1 score, while the $dLDA$ approach achieves the highest $bacc$ based on the average performance across all subjects. However, according to the results of the t -test, these two approaches are not significantly different from each other in the offline experiment setting. In the next section, we use these two approaches to conduct HITL experiments to compare their performance further. Note that the number of selected features varies significantly between individual subjects. This may be due to unique patterns of muscle and kinematic activity exhibited by each subject when moving their arms, potential difficulties encountered by the sensors during data collection, or limitations in the brute-force search method used to find the optimal combination of features and clusters. While the exact cause remains uncertain, we acknowledge this as an area for further exploration.

¹⁰The t -tests use the Šidák correction at a 95% confidence level.

Table 8: Average F_1 score, balanced accuracy (*bacc*), and task completion time for subjects using the $PM_{classifier}$ and $dLDA$ approaches in the HITL experiments.

Subject ID	F_1 score		<i>bacc</i>		Time	
	$PM_{classifier}$	$dLDA$	$PM_{classifier}$	$dLDA$	$PM_{classifier}$	$dLDA$
11	0.353	0.246	0.630	0.569	9.348	10.793
12	0.287	0.252	0.590	0.573	8.872	8.078
13	0.433	0.286	0.678	0.592	10.948	11.316
14	0.490	0.375	0.709	0.643	8.577	8.778
15	0.559	0.254	0.751	0.573	11.729	25.084
16	0.404	0.271	0.662	0.583	6.986	7.616
Average	0.421	0.281	0.670	0.589	9.410	11.944
Std. dev.	0.097	0.049	0.057	0.028	3.656	21.114
<i>p</i> -value	0.010		0.011		0.196	

5.4 Human-In-The-Loop Experiments

The human-in-the-loop (HITL) experiments involved six additional subjects (subject IDs 11 to 16, none of whom had prior experience with our designed experiment), performing the Refined Clothespin Relocation Task (RCRT) as detailed in Section 2.2.2. The HITL experiment consists of two phases: the training phase and the testing phase.

Training phase: During training, all sensors were activated to collect signals with 59 features, while the virtual avatar (Fig. 1c) was visible in the VR environment, mirroring the real arm’s movements. The wrist of the subject was constrained using a brace, only allowing forearm rotation movement. Subjects were instructed to complete all tasks across ten iterations, with each iteration consisting of eight distinct goals performed only once. All signals collected during this phase were used to train the two best-performing approaches identified in the offline experiment, namely $PM_{classifier}$ and $dLDA$, as we aimed to compare these approaches in the HITL experiment settings.

Testing phase: During this phase, subjects were requested to use the two trained approaches to control the virtual prosthesis, see Fig. 2a. For each approach, subjects were instructed to complete five iterations of the same task as they performed during the training phase. However, in the testing phase, kinematic sensors below the elbow joint were deactivated to simulate scenarios of patients with disability. We observed and collected data to assess whether the “simulated disabled patients” could successfully complete the tasks supported by GR techniques. Note that the subjects tested the $PM_{classifier}$ and $dLDA$ approaches in random order; half of the cohort began with $PM_{classifier}$, while the rest started with $dLDA$. This way, we intended to minimize any learning effects where experience gained from the first approach could improve performance in the second, ensuring a fair comparison. To assess performance, we used the same performance measures: the F_1 score and balanced accuracy. Additionally, we measured the average time spent by subjects picking up each clothespin and relocating it using the provided approaches. Note that during HITL experiments, we performed alignment weight computations in parallel to enhance reaction speeds. While the offline experiments were run on a cloud server, the HITL experiments were performed on a lab PC with an Intel® Core™ i7-8700K processor at 3.7GHz.

Table 8 summarizes the performance of the $PM_{classifier}$ $dLDA$ techniques in the

HITL experiments. It presents the F_1 score, $bacc$, and the average task completion time for the six subjects (subject IDs from 11 to 16) achieved by the compared techniques. The third-to-last row, labelled “Average” lists the average F_1 score, $bacc$, and completion time across the six subjects for each approach, $PM_{classifier}$ and $dLDA$. The second-to-last row, labeled “Std. dev.,” displays the standard deviation for all the measures across the subjects. The last row, labeled “ p -value,” presents the p -values from the t -tests comparing the average performance of the two approaches, measured for the F_1 score, $bacc$, and completion time. The results demonstrate that, on average, across all subjects, $PM_{classifier}$ outperforms the $dLDA$ baseline significantly in terms of the F_1 score and $bacc$. Furthermore, subjects utilizing the prosthesis controlled by the $PM_{classifier}$ approach completed tasks faster, on average, with a smaller standard deviation. This suggests that the $PM_{classifier}$ approach is not only quicker but also less sensitive to human variations compared to the $dLDA$ baseline. These findings underscore the potential of the $PM_{classifier}$ method in the domain of prosthetics.

6 Discussion

In this work, we present a novel approach for target pose recognition using PM-based GR to guide powered transhumeral prostheses. Evaluation results demonstrate that this approach achieves higher goal inference accuracy compared to baseline methods. Furthermore, HITL experiments confirm the feasibility of applying this approach in real-world settings, enabling interaction with humans to complete tasks.

Process mining techniques focus on analyzing data sequences with discrete values, which is a significant limitation because many real-world scenarios involve sequential data with continuous values, such as the prosthetic scenario studied in this work. This work explores the potential of continuous signal processing by aiming to transform continuous EMG signals into discrete classes, thereby enabling process mining techniques to handle continuous data effectively. The central idea in this line of work involves performing classification before applying process mining. In our previous work, the K -means clustering was used in offline settings because this technique focuses on classifying data points without learning criteria for classification. In contrast, this work introduces the LDA classifier, which learns pre-trained classification criteria, making it suitable for real-time applications rather than being limited to offline use.

The dynamic LDA ($dLDA$) introduced in Section 4.2 is an improvement over existing work on LDA classifiers applied to prosthetic scenarios [38]. In existing works, the classifier is trained using only final position data and does not utilize the data points collected during movement, which we refer to as static LDA ($sLDA$). In our conference paper [32], we introduced the PM-based GR approach with K -means clustering and demonstrated that it outperformed existing methods, such as $sLDA$ and $LSTM$. With the improved LDA classifier, $dLDA$, we use offline experiments in this paper to show that $dLDA$ outperforms $sLDA$, $LSTM$, and even PM-based GR with K -means clustering. These results highlight the significance of this improvement.

The main contribution of this work, however, is the new PM-based GR approach with an LDA classifier ($PM_{classifier}$), which is designed to address challenges in the HITL setting. When evaluating this new approach, we compared it to the $dLDA$ clas-

sifier, which is the best baseline according to the offline experiment. In the offline evaluation, see Table 5, $PM_{classifier}$ generally shows a monotonically increasing trend in both F_1 score and balanced accuracy as more data is collected by the sensors. In contrast, $dLDA$ can experience a decrease in performance at certain positions on the way to the target pose. As a result, $PM_{classifier}$ typically outperforms $dLDA$ as the movement trajectory approaches the target, which is desirable in prosthetic applications, as the target limb pose is crucial for task completion, such as grasping an object.

The results of the HITL experiments demonstrate that $PM_{classifier}$ outperforms $dLDA$, providing evidence of its value and making a substantial contribution to the field of prosthesis development. $PM_{classifier}$ demonstrated more consistent completion times compared to $dLDA$. $PM_{classifier}$ also achieved significantly better GR performance in terms of the F_1 score and balanced accuracy ($p < 0.05$), see Table 8. These findings highlight the robustness of the $PM_{classifier}$ approach against human variations in input signals under HITL conditions, making it well-suited for real-life applications. However, the GR performance in HITL settings was significantly lower than that observed in the offline experiments, which is commonly reported in the literature [10]. This highlights the inherent challenges of real-time HITL scenarios, which could be addressed in future work.

Although this work introduces a novel and promising approach for target pose recognition aimed at guiding powered transhumeral prostheses, several aspects could be further elaborated. The current feature selection and event discretization approaches, while effective, leave room for improvement. In future work, we are eager to explore other feature selection and event discretization techniques. For example, the brute-force search method, used to determine which features to select and how many clusters to use for grouping original signals into events, could be replaced with efficient heuristics. In addition, instead of relying on the LDA classifier, one can explore the benefits of using other machine learning-based classifiers in the PM-based GR approach. Furthermore, the current method for partitioning the trajectory and labeling data points could be improved, as there is no clear criterion for determining the number of segments. In this paper, we use five segments for the analysis. Exploring more effective ways to label data points for training classifiers remains an area for future investigation. It is interesting to identify feature conditions that describe *meaningful* prosthetic postures, and test whether specific sequences of these yield good predictors for the goals being pursued. Doing so will support one of the key advantages of our PM-based GR approach, namely, explanations of the obtained goal inferences based on the commonalities and discrepancies between the process models and the fresh observations on the level of meaningful patterns of events.

Yet another area for improvement is alignment computation. While the process discovery technique we used runs in linear time relative to the size of the input event log [18], the employed alignment technique is in the worst case exponential in the size of the solution [33]. The practical implications of the complexity of computing alignments arise when the model describes a large number of reachable states. In such cases, conformance checking becomes time-consuming. As the number of reachable states increases, the number of possible alignments to examine and select an optimal one grows exponentially. For HITL experiments, as goal inferences are required in (close to) real-time, alignment computation can become a burden. However, if the

model is not complex, that is, describes a small number of traces, the optimal alignment can be computed fast. We observed that discovered models are usually not overly complex, and hence, the time required to compute optimal alignments is manageable. For instance, the HITL experiments demonstrate that our approach can perform faster and more accurately than the baselines. However, addressing the worst-case scenarios is necessary if considering the technology for production. To this end, one can study the benefits of using alignment techniques designed for *online* conformance checking [35] or such that *approximate* optimal alignments fast [34]. We conjecture that approximations of optimal alignments computed in the time that is low polynomial in the sizes of the inputs can yield a good compromise of goal inference accuracy and runtime guarantees. An additional runtime improvement can stem from computing alignments for different process models in parallel.

The datasets for both offline and HITL experiments are arguably small, with the offline experiment using an existing dataset of 10 subjects and the HITL experiments including 6 subjects. The positive outcomes achieved so far encourage expanding the study to include a larger number of subjects to confirm the results' generalizability. Another aspect worth further exploration is the significant performance gap between the offline experiments and the HITL experiments. This discrepancy might be due to the dataset capturing features from forward-reaching tasks using *sound* limbs rather than from real-time control of a prosthesis. The difference between real-time prosthetic states in the HITL experiments and sound limbs introduces variations in feedback, which could affect the feature patterns collected during movement. These feature patterns can have an impact on the real-time recognition accuracy of machine-learning-based techniques [24, 36]. Therefore, collecting data from real-time control experiments and pre-defining standard traces toward the goal with distinguishable feature patterns for training the PM-based GR system has the potential to improve accuracy. Recent research suggests that when GR performance falls short of expectations, an adaptive GR system can automatically adjust its model to better fit and perform in current scenarios [31]. Using an adaptive GR technique, one could start by learning GR models from the movements of sound limbs. Then, as subjects collaborate with the prostheses to perform tasks, the adaptive GR technique can adjust the initially learned models to improve the HITL recognition performance.

This work highlights the potential of integrating process mining and goal recognition techniques into prosthetic control systems. By leveraging process models, we can not only improve recognition accuracy but also provide a more interpretable framework for understanding and explaining prosthetic behavior. This aligns with recent trends in adaptive GR systems [31], which aim to adjust models automatically to fit real-world scenarios better. As the field moves toward more personalized and adaptive prosthetic systems, our approach offers a promising foundation for future research.

7 Conclusion

This article presents a novel approach for using continuous, real-valued, multi-dimensional sensor data that characterizes the behavior of an observed autonomous agent to infer their goal using an existing GR system grounded in process mining techniques.

We evaluate the new approach in a study that aims to recognize the target poses of patients who use powered transhumeral prostheses. In this setting, the data from surface electromyography electrodes and kinematic sensors attached to patients is used to infer their intended poses and, subsequently, to guide powered prostheses in supporting patients' movements. In addition, we present a new linear discriminant analysis (LDA) classifier for recognizing target poses trained on the data gathered during the dynamic movement of the patients (dynamic LDA) that enhances the state-of-the-art LDA approach trained on the data collected during the fixed static patient's poses (static LDA). We use these LDA classifiers to conduct offline and human-in-the-loop experiments. The results of the experiments demonstrate that GR that relies on LDA classifiers for event discretization significantly outperforms state-of-the-art baselines.

References

- [1] Stefano V. Albrecht, Cillian Brewitt, John Wilhelm, Balint Gyevar, Francisco Eiras, Mihai Dobre, and Subramanian Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving. In *ICRA*, pages 1043–1049. IEEE, 2021.
- [2] Nasser A Alshamary, Student Member, Daniel A Bennett, Student Member, and Michael Goldfarb. Synergistic elbow control for a myoelectric transhumeral prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26:468–476, 2018.
- [3] Cillian Brewitt, Balint Gyevar, Samuel Garcin, and Stefano V. Albrecht. GRIT: Fast, interpretable, and verifiable goal recognition with learned decision trees for autonomous driving. In *IROS*, pages 1023–1030. IEEE, 2021.
- [4] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *International Conference on Pattern Recognition*, pages 3121–3124. IEEE Computer Society, 2010.
- [5] Peter Christen, David J. Hand, and Nishadi Kirielle. A review of the F-measure: Its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 56(3):73:1–73:24, 2024.
- [6] William H. E. Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1:7–24, 1984.
- [7] Luísa R. de A. Santos, Felipe Meneguzzi, Ramon Fraga Pereira, and André Grahl Pereira. An LP-based approach for goal recognition as planning. In *AAAI*, pages 11939–11946. AAAI Press, 2021.
- [8] Yiannis Demiris. Prediction of intent in robotics and multi-agent systems. *Cognitive Processing*, 8:151–158, 2007.
- [9] Purushothaman Geethanjali. Myoelectric control of prosthetic hands: State-of-the-art review. *Medical Devices: Evidence and Research*, 9:247–255, 2016.
- [10] Levi J. Hargrove, Erik J. Scheme, Kevin B. Englehart, and Bernard S. Hudgins. Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(1):49–57, 2010.
- [11] John A. Hartigan and Manchek A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [12] Jiayuan He, Xinjun Sheng, Xiangyang Zhu, Chaozhe Jiang, and Ning Jiang. Spatial information enhances myoelectric control performance with only two channels. *IEEE Transactions on Industrial Informatics*, 15(2):1226–1233, 2019.
- [13] Jin Huang, Guoxin Li, Hang Su, and Zhijun Li. Development and continuous control of an

- intelligent upper-limb neuroprosthesis for reach and grasp motions using biological signals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(6):3431–3441, 2021.
- [14] Ali Hussaini and Peter Kyberd. Refined clothespin relocation test and assessment of motion. *Prosthetics and Orthotics International*, 41(3):294–302, 2017.
- [15] Andres G. Jaramillo-Yanez, Marco E. Benalcázar, Sebastian Sardiña, and Fabio Zambetta. Towards discriminant analysis classifiers using online active learning via myoelectric interfaces. In *AAAI*, pages 6996–7004. AAAI Press, 2022.
- [16] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *AAAI*, pages 32–37. Morgan Kaufmann, 1986.
- [17] Anagha Kulkarni, Sarath Sreedharan, Sarah Keren, Tathagata Chakraborti, David E. Smith, and Subbarao Kambhampati. Designing environments conducive to interpretable robot behavior. In *IROS*, pages 10982–10989. IEEE, 2020.
- [18] Sander J. J. Leemans, Erik Poppe, and Moe Thandar Wynn. Directly follows-based process mining: Exploration & a case study. In *ICPM*, pages 25–32. IEEE, 2019.
- [19] Mathilde Legrand, Charlotte Marchand, Florian Richer, Amelie Touillet, Noel Martinet, Jean Paysant, Guillaume Morel, and Nathanael Jarrasse. Simultaneous control of 2DOF upper-limb prosthesis with body compensations-based control: A multiple cases study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 30:1745–1754, 2022.
- [20] Jianwei Liu, Xinjun Sheng, Dingguo Zhang, Jiayuan He, and Xiangyang Zhu. Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation. *IEEE Journal of Biomedical Health Informatics*, 20(1):166–176, 2016.
- [21] Bo Lv, Guohong Chai, Xinjun Sheng, Han Ding, and Xiangyang Zhu. Evaluating user and machine learning in short- and long-term pattern recognition-based myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:777–785, 2021.
- [22] Wookhee Min, Bradford Mott, Jonathan Rowe, Barry Liu, and James Lester. Player goal recognition in open-world digital games with long short-term memory networks. In *IJCAI*, pages 2590–2596. IJCAI/AAAI Press Press, 2016.
- [23] Reuth Mirsky, Sarah Keren, and Christopher Geib. *Introduction to Symbolic Plan and Goal Recognition*. Springer, 2021.
- [24] Max Ortiz-Catalan, Faezeh Rouhani, Rickard Brånemark, and Bo Håkansson. Offline accuracy: A potentially misleading metric in myoelectric pattern recognition for prosthetic control. In *EMBC*, pages 1140–1143. IEEE, 2015.
- [25] Nawadita Parajuli, Neethu Sreenivasan, Paolo Bifulco, Mario Cesarelli, Sergio Savino, Vincenzo Niola, Daniele Esposito, Tara J. Hamilton, Ganesh R. Naik, Upul Gunawardana, and Gaetano D. Gargiulo. Real-time EMG based pattern recognition control for hand prostheses: A review on existing methods, challenges and future implementation. *Sensors*, 19(20):4596, 2019.
- [26] Artem Polyvyanyy, Zihang Su, Nir Lipovetzky, and Sebastian Sardiña. Goal recognition using off-the-shelf process mining techniques. In *AAMAS*, pages 1072–1080. IFAAMAS, 2020.
- [27] Miquel Ramírez and Hector Geffner. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI*, pages 1121–1126. AAAI Press, 2010.
- [28] Charles Rich, Candace L. Sidner, and Neal Lesh. COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22(4):15–26, 2001.
- [29] Ahmed W. Shehata, Heather E. Williams, Jacqueline S. Hebert, and Patrick M. Pilarski. Machine learning for the control of prosthetic arms: Using electromyographic signals for improved performance. *IEEE Signal Processing Magazine*, 38(4):46–53, 2021.
- [30] Zihang Su, Artem Polyvyanyy, Nir Lipovetzky, Sebastian Sardiña, and Nick van Beest. Fast and accurate data-driven goal recognition using process mining techniques. *Artificial*

- Intelligence*, 323:103973, 2023.
- [31] Zihang Su, Artem Polyvyanyy, Nir Lipovetzky, Sebastian Sardiña, and Nick van Beest. Adaptive goal recognition using process mining techniques. *Engineering Applications of Artificial Intelligence*, 133:108189, 2024.
 - [32] Zihang Su, Tianshi Yu, Nir Lipovetzky, Alireza Mohammadi, Denny Oetomo, Artem Polyvyanyy, Sebastian Sardiña, Ying Tan, and Nick van Beest. Data-driven goal recognition in transhumeral prostheses using process mining techniques. In *ICPM*, pages 25–32, 2023.
 - [33] Wil van der Aalst, Arya Adriansyah, and Boudewijn van Dongen. Replaying history on process models for conformance checking and performance analysis. *WIREs Data Mining and Knowledge Discovery*, 2:182–192, 2012.
 - [34] Boudewijn van Dongen, Josep Carmona, Thomas Chatain, and Farbod Taymouri. Aligning modeled and observed behavior: A compromise between computation complexity and quality. In *CAiSE*, pages 94–109. Springer, 2017.
 - [35] Sebastiaan van Zelst, Alfredo Bolt, Marwan Hassani, Boudewijn van Dongen, and Wil van der Aalst. Online conformance checking: Relating event streams to process models using prefix-alignments. *International Journal of Data Science and Analytics*, 8:269–284, 2019.
 - [36] Richard B. Woodward and Levi J. Hargrove. Adapting myoelectric control in real-time using a virtual environment. *Journal of NeuroEngineering and Rehabilitation*, 16(11):1–12, 2019.
 - [37] Tianshi Yu, Ricardo Garcia-Rosas, Alireza Mohammadi, Ying Tan, Peter Choong, and Denny Oetomo. Comparing the outcomes of population-averaged and personalised input feature selection for transhumeral prosthetic interfaces. In *SMC*, pages 417–422. IEEE, 2021.
 - [38] Tianshi Yu, Alireza Mohammadi, Ying Tan, Peter Choong, and Denny Oetomo. Sensor selection with composite features in identifying user-intended poses for human-prosthetic interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:1732–1742, 2023.