Hierarchical Human-Motion Prediction and Logic-Geometric Programming for Minimal Interference Human-Robot Tasks

An T. Le¹ *, Philipp Kratzer¹, Simon Hagenmayer¹, Marc Toussaint²,³ and Jim Mainprice¹,²
*an.thai.le97@gmail.com

¹Machine Learning and Robotics Lab, University of Stuttgart, Germany
²Max Planck Institute for Intelligent Systems; IS-MPI; Tübingen/Stuttgart, Germany
³Technische Universität Berlin; TUB; Germany

Abstract—In this paper, we tackle the problem of human-robot coordination in sequences of manipulation tasks. Our approach integrates hierarchical human motion prediction with Task and Motion Planning (TAMP). We first devise a hierarchical motion prediction approach by combining Inverse Reinforcement Learning and short-term motion prediction using a Recurrent Neural Network. In a second step, we propose a dynamic version of the TAMP algorithm Logic-Geometric Programming (LGP) [1]. Our version of Dynamic LGP, replans periodically to handle the mismatch between the human motion prediction and the actual human behavior. We assess the efficacy of the approach by training the prediction algorithms and testing the framework on the publicly available MoGaze dataset [2].

I. INTRODUCTION

As robots become more capable, they will increasingly share space with humans. Consider the case where humans and robots additionally share a task, such as caring for people in a hospital. Robots could tidy, clean, bring lunch to patients, gather medication, or even prepare an operation room. The robot would be collaborating with one or several humans, not only sharing space but also partially executing the task the human is busy with. In such cases the user is interested in having maximal support from the robot while requiring a minimal amount of interference with its own task objectives. Humans do this naturally when they collaborate. For instance, one could put food back into the fridge while the other collects and cleans the dishes. In this paper we propose a framework that can produce such behaviors. We do this by combining hierarchical human motion-prediction and Task and Motion Planning (TAMP) framed as a Logic-Geometric Program (LGP) [1].

For safe and efficient human-robot collaboration it is crucial to account explicitly for the human when generating the robot behavior [3], [4]. Moreover, it is important to predict the human behavior in order to coordinate the human and robot actions, or interact without disturbing the natural flow of the human’s motion. This has been noted in numerous prior works [5], [6]. Hence, human intent and motion prediction is becoming an increasingly important topic of robotics research, which has been recently surveyed by Rudenko et al. [7].

A key challenge for human-motion prediction is to predict movement over a long horizon. This problem arises naturally in the case of sequences of manipulation motions, for example, when a human and robot have to prepare a table for dinner or tidy a room (see Figure 1). In this work, we propose a hierarchical prediction approach, which can handle such long-term horizon. The general approach is to use two hierarchical levels: symbolic, (i.e. discrete) and geometric (i.e., continuous). For the robot motion planning, we also use these two levels in a TAMP algorithm.

As human motion is the result of complex biomechanical processes that are challenging to model, state-of-the-art work on motion prediction focuses on data-driven models, such as recurrent neural networks [8], [9], [10]. Our prediction approach makes use of Maximum Entropy Inverse Reinforcement Learning [11], to produce a discrete policy, this policy decides what actions to take next based on the symbolic states of the world. At the lowest level we predict movements using a Recurrent Neural Network (RNN), which is trained on the MoGaze dataset [2]. In order to adapt the discrete policy to continuous motion prediction, we introduce goal conditioning to the RNN VRED architecture [8]. We combine two networks, one conditioned on hand target goals for manipulation and another one on pelvis target goals for walking. The high-level policy sequences goals by means of an intermediate grounding of symbolic layer, resulting in a
given this prediction, the planning module makes use of LGP to produce minimally interfering plans which support the human with the task objectives. There we devise a symbolic representation of the workspace and the task goal using the Planning Domain Definition Language (PDDL) [12]. Given both the geometric and the symbolic representation, LGP explores the space of skeletons using tree search, which results in a heuristic ranking of symbolic plans. Skeletons are then evaluated with increasing level of accuracy until a feasible plan is found. At the lowest-level a full trajectory is planned using non-linear programming. The optimizer uses an interior point method and a “Gauss-Newton” approximation of the Hessian [13], [14].

We combine human motion prediction and robot motion planning by predicting changes in the symbolic state and planning in the combined symbolic state. We first predict what the human would do and plan for the actions that the human would do last. In order to handle erroneous human motion prediction, we extend the basic formulation of LGP to handle dynamic changes in the workspace.

To summarize the main contributions of the paper:

• We propose a new formulation to produce long-term task sequences for a human-robot team that support the human while minimizing the interference.
• We introduce a new hierarchical motion prediction system which is able to produce full-body prediction in long horizons.
• We present results assessing the efficacy of our approach using the MoGaze [2] dataset.

This paper is organized as follows: In Section II we discuss relevant prior work. Section III introduces our framework, and gives information concerning our implementation. In Section V we evaluate our framework on motion capture data. Conclusions are drawn in Section VI.

II. RELATED WORK

A. Human-Robot Collaboration

Human-Robot Collaboration (HRC) focuses on robotic systems able to perform joint actions with humans [15], [16]. The robot is a member of a mixed human-robot team, where members share a common goal. In this context, shared task planning and interactive motion planning allow for higher level collaborations and is thus a topic of interest in HRC.

In order to schedule coordinated actions, a lot of work has explored how to model the capabilities of the agents in the workspace [17]. For intelligently account for space-sharing conventions the notion of Proxemics is now well accepted in HRC [18].

Some works include high-level symbolic planning in order to find a human-aware robot plan [19], [20] and also combining task and motion planning has shown success for human-aware HRC [21], [22]. However, no work proposes to integrate a full hierarchical predictive model of human behavior.

B. Human-Motion Prediction

Human movement is the result of simultaneous control of hundreds of degrees of freedom. However, muscles are controlled in coordination (i.e., synergies), which yields a low dimensional embedding of motion. Thus, time series techniques such as HMMs, though limited to low dimensional state spaces, have had some success in predicting movement and activities [23].

A lot of work in the area of movement prediction and regression of motion capture data focuses on data driven methods. In these applications less attention is put to reproducing physically correct forces, torques or muscle activation, but rather reproduce plausible movement. Nonlinear function approximators such as Gaussian Processes [24] or Deep Neural Networks [9], [25] have been used to regress large databases of human movement. Recurrent Neural Networks (RNN) are state of the art for predicting short-term high dimensional movements [8], [9], [26].

However, these methods mostly consider very short motions, often less than a second, and therefore are not suitable for predicting long sequences of manipulation tasks. To address this issue, we adapt a RNN architecture and make it goal-conditioned. We use a discrete, high-level policy in order to predict intermediate goals allowing us to predict long motion sequences by chaining the short-term predictions.

C. Task and Motion Planning

In classical AI, High Level Planning [27] has been studied for decades and many languages and planning paradigms have been developed to solve symbolic planning.

Research in Robotics has sought to integrate such concepts for the purpose of solving motion planning problems involving sequences of tasks such as object manipulation or footsteps. Task and Motion Planning (TAMP) is a subfield of motion planning and robotics that aims to find multiple intricate and sequential manipulation movements. Usually TAMP involves reasoning on a symbolic level, which provides discrete action sequences, and continuous motion planning, which tries to find motion trajectories fulfilling the discrete action sequence.

Approaches to TAMP often are random sampling methods, [28], [29], [30], constrained-based methods [31], [32] or numerical optimization based methods [1].

In this work we focus on Logic Geometric Programming (LGP) [1], which is an approach combining logic tree search with trajectory optimization techniques, and combine it with human motion prediction to plan a collaborative task in a scenario with a human-robot team. The uncertainty in human motion prediction is handled by dynamic replanning, which also extends the basic formulation of LGP to handle general dynamic workspaces. In [33], a human-robot collaboration task is implemented using LGP, where the human prediction is modeled with simple cost terms, and no replanning is performed. To our knowledge, LGP has never been combined with a learned predictive model of human motion.
III. HIERARCHICAL MOTION PREDICTION & PLANNING

Here we devise the framework for TAMP using a long-term prediction of human motion (see Figure 2). We rely on a symbolic decomposition to describe the task, which we encode using the Planning Domain Definition Language (PDDL) [12].

A. Dynamic LGP

For motion planning, we introduce Dynamic LGP, which is a variant of LGP, that has a replanning ability at Level 3 LGP [1], which given the current environmental conditions, solves for the minimal interference Human-Robot tasks.

The basic idea of LGP, is to decompose the task with two levels of abstraction. At the highest level we consider a discrete set of actions $A = \{a_i\}_{i=1}^{N}$, for instance move, pick and place (see Figure 4). We call a skeleton, a sequence of symbolic actions $a_{i:k}$. Here we denote $(R)$ to refer to the robot sequence of actions, as it is different from the human sequence of actions $a_{i:k}$ used in High-level Policy (see Figure 2).

A fully instantiated plan is then a skeleton, together with a motion trajectory $x : [0, T] \rightarrow \chi$, where $\chi = \mathcal{C} \times \mathcal{H} \times \mathcal{O}$, the Cartesian product of the robot, human and movable object configuration spaces respectively.

1) Problem formulation: An instance $I$ of Dynamic LGP consists of the following components:

Symbolic Domain:
- Predicates $\mathbb{P} = \{P_1(\cdot),...,P_N(\cdot)\}$
- Constants $O$ as terms/arguments for predicates $\mathbb{P}$
- All symbolic states $s \in \mathbb{S}$ in the domain, where each state is a set of grounded propositions from the predicates $\mathbb{P}$
- A set of actions $a = (R, P, E) \in A$ where:
  - $R$ : parameters of the action.
  - $P \subseteq \mathbb{P}$ : preconditions predicates.
  - $E \subseteq \mathbb{P}$ : effect predicates.

In our experiments, we adopt a PDDL-syntax to describe the symbolic domain.

Geometric problem: Let $\mathcal{C}$ be the configuration space of the robot and the geometric state at time $t$, $x_t \in \mathcal{C}$. The task is to find a global path $x : t \mapsto x_t$, which minimizes the following LGP:

$$
\min_{x, a_{i:K}} \int_0^{KT} c(x(t), \dot{x}(t), \ddot{x}(t), s_k(t))dt
$$

s.t.

$$(x(0) = x_0, h_{goal}(x(KT)) = 0, g_{goal}(x(KT)) \leq 0)$$

$$\forall t \in [0, KT] : h_p(x(t), \dot{x}(t), s_k(t)) = 0,$$ $g_p(x(t), \dot{x}(t), s_k(t)) \leq 0$

$$\forall k \in \{1,...,K\} : h_{sw}(x(t), \dot{x}(t), a_k) = 0$$

$$s_k \in \mathbb{S}_{goal}$$

where the path is global continuous $x$ and contains $K \in N$ phases, each has fixed duration $T > 0$.

In our experiments, the cost function $c : (q_t, \dot{q}_t, \ddot{q}_t, s) \mapsto c_t \in \mathbb{R}$, is a combination of differentiable maps, penalizing velocities and accelerations of the robot. Obstacle avoidance and goal manifold are enforced using equality and inequality constraints $h_p, g_p$ in the phase $k(t) \in [t/T]$ conditioned on a discrete symbolic state $s_k \in \mathbb{S}$.

To impose transition conditions between phases, the switch functions $h_{sw}, g_{sw}$ define equalities and inequalities constraints conditioned on the transition action $a_t$. We assume that the equality and inequality functions are differentiable.

2) Solving LGP: To search the symbolic domain for a skeleton satisfying all constraints, the action set $A$ has to be grounded with the constants set $O$ [34], resulting in the grounded action set $A_g$.

The most basic operations for searching are the feasibility check and the state transition. In this case, the operations can be formally stated as:

- Action feasibility check: A grounded action $a = (R, p, e) \in A_g$, in which $p, e$ are the grounded propositions of the preconditions and the effects [34], is applicable to $s$ iff $p \subseteq s$ with $\forall s \in \mathbb{S}$.
- State transition: new state $s' = \text{exec}_a(s) = s \cup e$ with $\forall s, s' \in \mathbb{S}$.

Fig. 2: Dataflow through our framework. The human motion prediction system produces a trajectory $\{h_1\}_{i=1}^N$ that is being used to deduce the future symbolic state of world. Every replanning event is triggered at $\tau$ where a Logic-Geometric Program is optimized.
Algorithm 1: Dynamic LGP

**input:** Init state \( x_0 \), goal set \( S_{\text{goal}} \)

Deduce symbolic state \( s_0 \) from \( x_0 \);

Search \( \Gamma_0(s_0, S_{\text{goal}}, I) \);

Set \( \kappa = a_{(R)}^{(H)} \in \Gamma_0 \) as best feasible skeleton;

Set elapsed time \( \tau = 0 \);

while \( S_{\text{goal}} \) not reached at current \( t \) do

Update system kinematics and human position;

Deduce current symbolic state \( s_t \) from \( x_t \);

if \( F(\kappa, x_t, s_t) = 0 \) then

Search \( \Gamma_t(s_t, S_{\text{goal}}, I) \);

Update \( \kappa = a_{(R)}^{(H)} \in \Gamma_t \);

Set elapsed time \( \tau = 0 \);

end

Optimize NLP (Level 3 in [1]) of \( \kappa \) from time \( \tau \);

Execute current action of the skeleton \( \kappa \);

\( \tau = \tau + 1 \);

Wait for next trigger;

end

where \( n(s) \) is the cardinality of the state, i.e. the number of grounded propositions. Using the heuristic, we search through the symbolic domain for all tie shortest solutions using Dijkstra algorithm.

Once all tie skeletons are found, we rank them by grounding them using simple interpolation paths and computing their costs defined in Equation (1a). We then solve the NLP instance in increasing cost order until a feasible solution is found.

To achieve human avoidance in single planning at the geometric level, we populate the human positions as obstacles along the human prediction trajectory. This ensures the worst-case scenario, in which the robot finds a collision-free trajectory at the beginning with single planning.

4) **Dynamic planning:** As the actual human behavior may deviate from the prediction, the motion trajectory or the skeleton \( a_{(R)}^{(H)} \) may become sub-optimal or even unfeasible.

Algorithm 1 describes the main execution of our Dynamic LGP formulation. The main idea is to enable the replanning capability for both: symbolic and geometric levels of LGP.

Initially, similar to single planning in Section III-A.3, the algorithm finds the best (i.e. lowest cost) feasible skeleton at the beginning and sets it to be the current executing skeleton \( \kappa \in \Gamma \).

For each replanning trigger, the algorithm validates for actual symbolic and geometric feasibility of the current executing skeleton. To validate the current executing skeleton, Dynamic LGP first deduces the current symbolic state \( s_t \) from the current geometric state \( x_t \). Recall that the geometric state \( x_t = (q_t, h_t, o_t) \), concatenates \( q_t \) is the robot, \( h_t \) the human, and \( o_t \) the movable object configurations. The current human state \( h_t \) is retrieved from the human trajectory \( \{h_t\}_{t=1}^{T} \) computed by Hierarchical Motion Prediction module (see Figure 2). Given the current human state \( h_t \), it also updates the collision avoidance inequality constraint \( g_p \)

| \( \text{Start State} \) | \( (0, 4, 0, 1, 0, 3, 1, 0, 1, 2) \) |
|---|---|
| \( \text{Actions} \) | Go to white shelf |
| | Pick up cup |
| | Go to table |
| | Place |
| \( \text{End State} \) | \( (1, 3, 0, 1, 0, 3, 1, 0, 1, 0) \) |

| \( \text{Predicate} \) | \( \text{Definition} \) |
|---|---|
| \( \text{on X Y} \) | check if exists a stable 3D \( xy\phi \) joint from X to Y |
| \( \text{at X Y} \) | check if \( \|x - x\|_2 \leq \tau \) from X to Y |
| \( \text{carry X Y} \) | check if exists a stable free joint (6D) from X to Y |

| \( \text{TABLE II: Example high-level trajectory} \) |

at the current phase \( k(t) \) in the NLP.

As an example of Symbolic Deduction, the predicate (on X Y) is deduced by checking in the system kinematic tree if there is a stable 3D \( xy\phi \) joint from X to Y. Table I describes our setup symbolic inference for the predicates using the system kinematics. Specifically, querying (human-carry, \( ?x \) - object) or (agent-carry, \( ?x \) - object) predicates can be done using (carry X Y) check. This is the mechanism to encode the human intention to the planner symbolically.

Then, the executing skeleton at the current time \( t \) can be checked for feasibility, both symbolically and geometrically. Formally, given the current symbolic and geometric state \( x_t, s_t \), the \( \kappa \) skeleton feasibility at the current time \( t \) is defined as:

\[
F(\kappa, x_t, s_t) = \begin{cases} 
1 & x_0 = x_t, s_0 = s_t \\
\exists x : [t, KT] \rightarrow C : (1b) - (1f) & \text{(4)} \\
0 & \text{otherwise}
\end{cases}
\]

If the skeleton \( \kappa \) is feasible, the NLP is optimized for \( \kappa \) from the elapsed time \( \tau \), i.e., \( \forall t \in [\tau, KT] \), given the current system kinematics condition. Otherwise, it discards the current executing skeleton and resets the elapsed time \( \tau = 0 \). The single planning is then triggered to replan a new skeleton solution set \( \Gamma_t(s_t, S_{\text{goal}}, I) \) and update the new best feasible skeleton. One may notice that the elapsed time \( \tau \) for the current executing skeleton is an implementation detail; however, it plays a crucial role in keeping track remaining execution time for the fixed duration phases of LGP.

**B. Long-Term Motion Prediction using Hierarchies**

At the top-level, our hierarchical motion prediction uses Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [11] and a low-level which performs full-body motion prediction conditioned on the sequence of sub-goals induced by the sequence of high-level actions \( a_{(R)}^{(H)} \) given by the high-level policy.

1) **Goal-Conditioning:** To be able to use motion prediction as a sub-policy, we do not only need a sequence-to-sequence mapping but also need it to be goal-conditioned.

Thus, we need a predictive function \( h_{t+1:T} = f(h_{0:t}, g^*) \) that computes a trajectory of future human states \( h_{t+1:T} \) given previous observed states \( h_{0:t} \) and a goal \( g^* \). We use VRED, a recurrent neural network-based model for
predicting motion [8] and make it goal-conditioned by adding a three-dimensional position \( g_t \) to the input of the network at every timestep (see Figure 3). The goal input \( g_t \) is relative to the coordinate frame of the human and thus changes every timestep.

Particularly, we train two networks, one conditioned on hand target goals for manipulation and another one on pelvis target goals for walking. The network is trained on full-body, kinematic motion trajectories. We use a mean squared distance loss between the base position and a quaternion that can solve a high-level task. Therefore, we use VRED due to the scaling property of deep models that learns high-dimensional configuration trajectory of human-motion captures.

2) High-level Policy Symbolic State Representation: We learn a policy \( \pi \) that can solve a high-level task. Therefore, we simplify the state-space to a symbolic representation and use tabular MaxEnt IRL to retrieve our policy. This makes the dataset well suited for our application.

MaxEnt IRL is based on state frequency calculations. For the MoGaze dataset (see Subsection IV), the discretized state is given by the number of objects on a location and the human position as follows:

\[
s(H) = (\text{cups - table}, \text{cups - shelf}1, \text{cups - shelf}2, \text{plates - table}, \text{plates - shelf}1, \text{jugbowl - table}, \text{jugbowl - shelf}1, \text{jugbowl - shelf}2, \text{humanPos}).
\]

The action space is discretized similarly. An example skeleton can be seen in Table II.

We use heuristics for interfering the exact goal for the human hand or pelvis, for example, by computing the closest point on the table to the human which is not occupied. The heuristics could be further improved by the use of human intention prediction as in [35].

The full long-term prediction is achieved by obtaining the skeleton from the high level policy \( \pi \), extracting the goals for the low-level from the heuristics according to the actions in the trajectory, and using the goal-conditioned RNN to obtain a sequence of full-body trajectories corresponding to the high-level actions.

IV. Dataset

We test our framework on the MoGaze dataset [2]. The dataset contains 180 minutes of long, full-body motion sequences for six humans, with 1627 pick and place actions being performed.

Besides human data, the dataset contains object data for two shelves, a table, and 10 movable objects like cups and plates. The participants performed simple manipulation tasks, such as setting up the table for a fixed number of persons or putting a set of specified objects onto one of the shelves. This makes the dataset well suited for our application.

V. Experiments

Here we report three types of experiments. First, we report results on hierarchical motion prediction. Then we report two sets of experiment using LGP, for coordinated human-robot tasks. In the first, we report a large scale experiment on 63 segments of setting up the table for 2-3 persons, where we use a degraded ground-truth trajectory as the prediction. In the second, we report results on the full pipeline using only 8 segments.

A. Long-Term Motion Prediction using Hierarchies

We first compare the original VRED implementation with the VRED conditioned on goal inputs on the MoGaze dataset. Results show that the goal-conditioned prediction network achieves both a better joint angular error of 7.99 instead of 10.14 and a significantly better base position error of 3.84 instead of 12.56, than the network without goal-conditioning. This is expected because the goal-conditioned network uses oracle information of the goal position. To test the accuracy of the high-level policy, we extracted the task of setting up the table for one person from the dataset. We then run tabular MaxEnt IRL, showed that the

```
(define (domain set_table)
  (:requirements :typing)
  (:types location object)
  (:constants
    table small_shelf big_shelf - location
    cup_red cup_green cup_blue cup_pick plate_pick plate_red plate_green
    plate_blue jug_bowl - object)
  (:predicates
    [agent-at ?l - location]
    [not ?x - object ?l - location]
    [agent-free]
    [agent-avoid-human]
    [agent-carry ?x - object]
    [human-carry ?x - object]
  )

  (:action place
    :parameters (?l - location)
    :precondition (and (agent-at ?l))
    :effect (and (not (on ?x ?l)) (not (agent-free)))
  )

  (:action pick
    :parameters (?x - object ?l - location)
    :precondition (and (agent-at ?l) (not (human-carry ?x)))
    :effect (and (not (on ?x ?l)) (not (agent-free)) (agent-carry ?x))
  )

  (:action move
    :parameters (?l - location)
    :precondition (not (agent-at ?x))
    :effect (and (not (agent-at ?*)) (agent-at ?l))
  )
)
```

Fig. 3: Structure of the VRED [8] with the goal input added.

Fig. 4: PDDL-syntax symbolic domain of set-table task.
initial states

get blue plate

get red plate

gate blue cup

gate red cup

Fig. 5: Dynamic LGP iterations using human ground truth. Note that in the bottom row images, the robot trajectory is green.

learned policy solved the task in 80% of the cases. However, a perfect imitation was achieved solely in 16% of the test runs of the cross-validation. This is because the algorithm is limited by our symbolic state and action representation. Including more complex state features, e.g., from the 3D environment, could further improve the algorithm.

We also observe that for the walking action the human collides with a chair, as the model is not aware of scene geometry. In future work this could be avoided by using a prediction method that accounts for the scene, such as [26].

B. Dynamic LGP with Human Ground Truth

To test Dynamic LGP, we design the PDDL-syntaxis domain following the available objects in the MoGaze dataset. Figure 4 presents the domain for a set-table task, which consists of a set of necessary predicates and a set of actions, in which the robot and the human cooperate to pick and place objects setting the dinner table for 2-3 persons.

We select 63 dataset segments for this task, and automate inferring the start symbolic state from the environment kinematics. We define the robot goal for each segment, e.g. $s_0 = \{(\text{agent-free}), (\text{agent-avoid-human}), (\text{on cup-green big-shelf}), (\text{on plate-blue small-shelf})\}$ and the goal state $s_g = \{(\text{on cup-green table}), (\text{on plate-blue table})\}$. In this experiment we directly feed the human trajectory ground truth into the Dynamic LGP. We randomly remove a part of the human trajectory in the dataset for each segment to simulate human prediction data.

The overall task Intersection over Union (IoU) between the set of objects the human and the robot must place on the table is 0.64 ± 0.30. In other words, most of the robot task instance has more than half of the objects to pick and place overlapping with the human task. Dynamic LGP needs to recognize the overlapping part and plan accordingly.

Fig. 6: Total time (left) and number of solved NLPs (right) to find an overall feasible solution over skeleton length.

Fig. 7: Total time (left) to find an overall feasible solution and skeleton length (right) over task progress.

The replanning trigger rate is set every 10 timesteps. Note that if the trigger rate is higher, the executing skeleton would be updated more frequently. Higher rates are desirable to handle more rapid changes in the environment however it has to be balanced with the computational capabilities of the robot. The move action has fixed duration of 30 timesteps, the pick and place actions both have fixed duration of 5 timesteps. The sampling rate is 10Hz.

We pause the simulation environment each time the planner is triggered so that the robot reacts in time to avoid the human obstacle. An example of a dynamic LGP run is demonstrated in Figure 5, where the task instance consists of 7 objects to pick and place. The robot executes pick and place for 4 objects, and the human executes the other 3 objects.
Our experiments show that Dynamic LGP is able to produce plans that have higher success rate than single planning. These plans also reduce the total time to execute the task.

VI. CONCLUSIONS & FUTURE WORK

We then run two planning modes; single planning and dynamic planning for each of the 63 segment. The task instance is considered successful if, at the end of the robot trajectory, the deduced symbolic state is in the goal set \( S_{goal} \). For dynamic planning, the task fails when the timeout for Algorithm 1 is reached while the goal set is not satisfied. For single planning, the task fails when no feasible skeleton is found.

Table III summarizes the statistics in terms of success rate, symbolic planning time, task time reduction (i.e., the original time taken by the human to perform the task in the dataset compared to the execution time with support from the robot) and path ratio (i.e., the ratio of distance traveled by the robot, with the distance of single planning is the baseline). Each category is reported in mean and standard deviation over all task instances. All experiments have been performed with an AMD Ryzen 7 5800X @ 3.8GHz.

As one can see the single planning success rates are lower than dynamic planning. This is expected since the single planning does not account for potential mismatch between the degraded ground truth and the actual ground truth.

Figure 8 depicts the recorded trajectories of both planning modes and the human trajectory in a task instance. We see that the dynamic mode plans a much shorter trajectory going between the shelf and the table because the only human obstacle constraint is efficiently updated at every trigger. We also see in Table III that the trajectory length of single planning is almost twice of dynamic planning (i.e., path ratio).

Surprisingly, the experiment shows that the executing symbolic skeleton in dynamic planning is usually invalid over task progress. Hence full LGP replanning is triggered frequently. This shows that the replanning capability is crucial in a dynamic environment, such as working with humans, since the symbolic state deduced from the environment is rapidly changed.

Figure 6 reports both the total solution time and the number of NLPs that have to be solved to find a feasible solution. As can be seen, most of the time only one NLP is needed to reach a feasible solution. This implies that usually for this task, the lowest cost skeleton is feasible. In other cases, more NLPs need to be solved due to the dynamic characteristic of the task.

Generally, the longer the action skeleton, the longer it takes to solve one NLP. The figure shows a median runtime of about 9 sec for the longest sequence length of 16, with \( \approx 450 \) time steps to optimize in an NLP. This time step correspond the time discretization of interior point trajectory optimization algorithm [13], [14]. Notice in Figure 7 that in some cases, the action skeleton length increases as the task progresses. The skeleton length’s median is 6 at task progress ratio 0.3, then increases to 8 at task progress ratio 0.4. This implies that the LGP replan sometimes has to resort to longer action skeletons with higher costs as shorter skeletons are infeasible.

C. Dynamic LGP with Long-Term Prediction

In this experiment, we choose 8 data segments from MoGaze, and produce the Long-Term Prediction outputs as described in Section III-B.

For each segment, we run 5 task instances to capture the human motion prediction statistics due to its stochasticity. The settings are the same as in Section V-B. The overall task IoU between the robot and the human objects is \( 0.34 \pm 0.13 \). Obviously, this IoU is less than Human Ground Truth experiment since in the Human Ground Truth experiment the human trajectory is used directly.

Table IV reports task statistics for this experiment. Statistics agree with Table III, which shows that dynamic planning has higher success rates and produces shorter paths and needs slightly more time to complete than single planning.

Figure 9 also shows decreasing total solution times as the task progresses with human prediction. Note that the planning time are an order of magnitude lower, as the task includes less objects.

### TABLE III: Dynamic LGP with Human Ground Truth

|                     | Single planning | Dynamic planning |
|---------------------|-----------------|------------------|
| Success rate        | 84.1\%          | 95.2\%           |
| Symbolic plan time  | 0.032 ± 0.036(\text{sec}) | 0.045 ± 0.053(\text{sec}) |
| Task time reduction | 0.577 ± 0.107   | 0.683 ± 0.099    |
| Path ratio          | 1.000           | 0.584 ± 0.148    |
| LGP replan count    | -               | 4.83 ± 2.21      |

### TABLE IV: Dynamic LGP with Human Prediction

|                     | Single planning | Dynamic planning |
|---------------------|-----------------|------------------|
| Success rate        | 91.2\%          | 100\%            |
| Symbolic plan time  | 0.00005 ± 0.00001(\text{sec}) | 0.00006 ± 0.00002(\text{sec}) |
| Task time reduction | 0.298 ± 0.078   | 0.300 ± 0.100    |
| Path ratio          | 1.000           | 0.026 ± 0.155    |
| LGP replan count    | -               | 3.0 ± 0.87       |

Fig. 8: Recorded actual trajectories of single planning (green), dynamic planning (blue) and actual human trajectory (red) on the workspace.
the task by a factor approaching 2, which is what one would expect when two agents collaborate at a task.

The dynamic version of the LGP algorithm we introduce replans periodically to handle the mismatch between the human motion prediction and the actual human motion behavior. Our experiments demonstrate that the solutions obtained with human prediction are more efficient and have higher success rate. We also show that the hierarchical motion prediction approach is capable to produce long-term human motion predictions. Furthermore, we design the integration between hierarchical human motion prediction with Dynamic LGP to apply in human-robot collaboration scenarios.

In future work, we aim to include collision avoidance in our hierarchical motion prediction framework and produce full-body robot motions using a Level 2 NLP for LGP (we refer the reader to [1]) to handle complete robot grasping configurations. We also plan to port these results to the Pepper robot of the University of Stuttgart.

ACKNOWLEDGMENT

This work is partially funded by the research alliance “System Mensch”. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Philipp Kratzer.

REFERENCES

[1] M. Toussaint, “Logic-geometric programming: An optimization-based approach to combined task and motion planning,” in IJCAI, 2015.
[2] P. Kratzer et al., “Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze,” IEEE Robotics and Automation Letters, vol. 6, no. 2, 2020.
[3] A. Bauer, D. Wollherr, and M. Buss, “Human–robot collaboration: a survey,” International Journal of Humanoid Robotics, vol. 5, no. 01, 2008.
[4] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, “Human-aware robot navigation: A survey,” Robotics and Autonomous Systems, vol. 61, no. 12, Dec. 2013.
[5] A. Walter, A. C. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, “Task planning for human-robot interaction,” in Proceedings of the joint conference on Smart objects and ambient intelligence, 2005.
[6] S. Lemaigre, M. Wariner, E. A. Sisbot, A. Clodic, and R. Alami, “Artificial cognition for social human–robot interaction: An implementation,” Artificial Intelligence, vol. 247, 2017.
[7] M. Gharbi, R. Lallement, and R. Alami, “Combining symbolic and geometric planning to synthesize human-aware plans - toward more efficient combined search,” IEEE/RSJ Int. Conf. on Intel. Robots And Systems (IROS), 2015.
[8] B. Busch, M. Toussaint, and M. L. 0001, “Planning Ergonomic Sequences of Actions in Human-Robot Interaction.” IEEE Int. Conf. Robotics And Automation (ICRA), 2018.
[9] D. Kulic et al., “Incremental learning of full body motion primitives and their sequencing through others observation,” The Int. Journal of Robotics Research, vol. 31, no. 3, 2012.
[10] P. Kratzer et al., “Towards combining motion optimization and data driven dynamical models for others prediction,” in IEEE-RAS Int. Conf. on Humanoid Robotics (Humanoids), 2018.
[11] K. Filipiak, S. Levine, P. Felsen, and J. Malik, “Recurrent neural networks for human dynamics,” in IEEE Int. Conf. on Computer Vision (ICCV), 2015.
[12] P. Kratzer, M. Toussaint, and J. Mainprice, “Prediction of human full-body movements with motion optimization and recurrent neural networks,” in IEEE Int. Conf. Robotics And Automation (ICRA), 2020.
[13] M. Gharbi, D. Nau, and P. Traverso, Automated Planning: theory and practice. Elsevier, 2004.
[14] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, “Manipulation planning with probabilistic roadmaps,” The Int. Journal of Robotics Research, vol. 23, no. 7-8, 2004.
[15] L. P. Kaelbling and T. Lomazo-Pérez, “Hierarchical task and motion planning in the now,” in IEEE Int. Conf. Robotics And Automation (ICRA), 2011.
[16] N. T. Dantam et al., “An incremental constraint-based framework for task and motion planning,” The Int. Journal of Robotics Research, vol. 37, no. 10, 2018.
[17] L. Lomozo-Pérez and L. P. Kaelbling, “A constraint-based method for solving sequential manipulation planning problems,” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014.
[18] F. Lagriffoul et al., “Efficiently combining task and motion planning using geometric constraints,” The Int. Journal of Robotics Research, vol. 33, no. 14, 2014.
[19] M. Toussaint and M. Lopes, “Multi-bound tree search for logic-geometric programming in cooperative manipulation domains,” in IEEE International Conference on Robotics and Automation (ICRA), 2017.
[20] M. Fox and D. Long, “Pddl2.1: An extension to pddl for expressing temporal planning domains,” Journal of Artificial Intelligence Research, vol. 20, Dec 2003.
[21] P. Kratzer et al., “Anticipating human intention for full-body motion prediction in object grasping and placing tasks,” in IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN), 2020.