Task and Motion Planning for Human-Robot Collaboration using Non-Linear Programming and Hierarchical Motion Prediction

An T. Le[1], Philipp Kratzer[1], Simon Hagenmayer[1], Marc Toussaint[2,3] and Jim Mainprice[1,2]
firstname.lastname@ipvs.uni-stuttgart.de
[1]Machine Learning and Robotics Lab, University of Stuttgart, Germany
[2]Max Planck Institute for Intelligent Systems ; IS-MPI ; Tübingen/Stuttgart, Germany
[3]Technische 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 of tidying a kitchen, where the human is interested in having maximal support from the robot while requiring a minimal amount of interference with its 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. The case falls into Human-Robot Collaboration (HRC) category, which focuses on robotic systems able to perform joint actions with humans [3], [4]. The robot is a member of a mixed human-robot team, where members share a common goal. Shared task planning and interactive motion planning allow for higher-level collaboration. The notion of *Proxemics* to intelligently account for space-sharing conventions is now well accepted in HRC [5]. To schedule coordinated actions, many works have explored how to model the capabilities of the agents in the workspace [6]. Some works include high-level symbolic planning in order to find a human-aware robot plan [7], [8] and also combining task and motion planning has shown success for human-aware HRC [9], [10]. However, no work proposes to integrate a full hierarchical predictive model of human behavior.

In this work, we propose a framework combining hierarchical human motion-prediction and Task and Motion Planning (TAMP) to tackle HRC tasks with human intention prediction. We propose a hierarchical prediction approach for long-term human motion prediction, which can handle such a long-term horizon. For TAMP, we focus on Logic Geometric Programming (LGP) [1], which is an approach combining logic tree search with trajectory optimization techniques, and adapt it to plan a collaborative task in a scenario with a human-robot team. We combine human motion prediction and robot motion planning by predicting changes in the symbolic state and planning in the combined symbolic state. To handle erroneous human motion prediction, we extend the basic formulation of LGP to handle dynamic changes in the workspace. In [11], 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. To summarize the main contributions of the paper: First, we propose a new formulation to produce long-term task sequences for a human-robot team that support the human while minimizing the interference. Second, we introduce a new hierarchical motion prediction system that can produce full-body prediction in long horizons. Finally, we present results assessing the efficacy of our approach using the MoGaze [2] dataset.

II. HIERARCHICAL MOTION PREDICTION & PLANNING

Here we devise the framework for TAMP using a long-term prediction of human motion.

A. Dynamic LGP

For motion planning, we introduce Dynamic LGP, which is a variant of LGP, to introduce the replanning ability to plan Level 3 LGP [1] at given current environmental conditions , solving 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 $A = \{a_i\}_{N}^{i=1}$, for instance move, pick and place (see Figure 2). We call a *skeleton*, a sequence of symbolic actions $a_{1:K}$ 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:

- A set of predicates $\mathbb{P} = \{P_1(.), ..., P_N(.)\}$.
- A set of constants O as terms/arguments for predicates $\mathbb{P}$.

- A set of 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 A of actions $a = (R, P_+, P_-, E_+, E_-)$ where:
  - R is the set of parameters of the action.
  - P_+, P_- are the sets of positive and negative pre-condition predicates $P_+, P_- \subset \mathbb{P}$.
  - E_+, E_- are the sets of positive and negative effect predicates $E_+, E_- \subset \mathbb{P}$.

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

$$\min_{x, a_{1:K}, s_{1:K}} \int_0^{KT} c(x(t), \dot{x}(t), \ddot{x}(t), s_{k(t)})\mathrm{dt} \tag{1a}$$

s.t.

$$x(0) = x_0,\ h_{\text{goal}}(x(KT)) = 0,\ g_{\text{goal}}(x(KT)) \leq 0 \tag{1b}$$

$$\forall t \in [0, KT] : h_p(x(t), \dot{x}(t), s_{k(t)}) = 0,$$
$$g_p(x(t), \dot{x}(t), s_{k(t)}) \leq 0 \tag{1c}$$

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

$$s_k \in \text{exec}_{a_k}(s_{k-1}) \tag{1e}$$

$$s_K \in \mathbb{S}_{\text{goal}} \tag{1f}$$

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_k. 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 [12], 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 $a = (R, p_+, p_-, e_+, e_-) \in A_g$, in which p_+, p_-, e_+, e_- are the positive and negative grounded propositions of the preconditions and the effects, is applicable to s iff $p_+^a \subset s$ and $p_-^a \cap s \in \emptyset$ with $\forall s \in \mathbb{S}$.
- **State transition**: new state $s' = \text{exec}_a(s) = (s \setminus e_-) \cup e_+$ with $\forall s, s' \in \mathbb{S}$.

For a given symbolic goal set $\mathbb{S}_{\text{goal}} \subset \mathbb{S}$, these two operations, allow us to instantiate a search process using any tree search algorithm (i.e., depth first, breadth first, etc). If a skeleton feasible skeleton $a_{t:K}$ leading to symbolic goal state $s^g \in \mathbb{S}_{\text{goal}}$ is found, a Non-Linear "trajectory optimization" Program (NLP) is defined. The NLP considers geometric switches in the system kinematics with long-term dependencies. In our implementation, we use an interior point method [13], [14] to optimize this NLP.

Algorithm 1: Dynamic LGP

```
input: Init state x_0, goal set S_goal
Infer symbolic state s_0 from x_0;
Search Γ_0(s_0, S_goal, I);
Set κ = a_{1:K_0} ∈ Γ_0 as best feasible skeleton;
Set elapsed time τ = 0;
while S_goal not reached at current t do
    Update system kinematics and human position;
    Infer current symbolic state s_t;
    if F(κ, s_t) = 0 then
        Search Γ_t(s_t, S_goal, I);
        Update κ = a_{1:K_t} ∈ Γ_t as best feasible
         skeleton;
        Set elapsed time τ = 0;
    end
    Optimize NLP (Level 3 in [1]) of κ from time τ;
    Execute current action of the skeleton κ;
    τ = τ + 1;
    Wait for next trigger;
end
```

3) Single planning: Given the dynamic LGP instance I and current symbolic state s_0, we define the set of all skeletons leading to $\mathbb{S}_{\text{goal}}$ as:

$$\Gamma(s_0, \mathbb{S}_{\text{goal}}, I) = \{a_{1:K} : \forall_{i=1}^K a_i \in A_g,$$
$$s_i = \text{exec}_{a_i}(s_{i-1}), s_K \in \mathbb{S}_{\text{goal}}\} \tag{2}$$

For search efficiency, we define a simple heuristic to guide the search as the symbolic distance from the current state and the goal. The distance is defined as:

$$h(s) = n(s_+^g \setminus s) + n(s_-^g \cup s) \tag{3}$$

where $n(s)$ is the cardinality of the state, i.e. the number of grounded propositions. s_+^g, s_-^g are the positive and negative proposition set of the goal state $s^g \in \mathbb{S}_{\text{goal}}$. Using the heuristic, we search through the symbolic domain for all tie shortest solutions using the 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 $a_{1:K}$ may become sub-optimal or even unfeasible. Hence a crucial component for dynamic LGP is to infer the current symbolic state from the current environmental condition. For example, the predicate (on X Y) is inferred 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

(on X Y)	check if exists a stable 3D $xy\phi$ joint from X to Y
(at X Y)	check if $\|x_X - x_Y\|_2 \leq r \| r \in \mathbb{R}$
(carry X Y)	check if exists a stable free joint (6D) from X to Y

TABLE I: Predicate inference

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 symbolic query is the primary mechanism encoding the human intention into the domain design. For example, in the set-table task, the proposition describes the object carried by the human (human-carry, object) defined to be in the goal set, assuming that the human intentions are always to cooperate to complete the task. Then the robot can plan the remaining actions to reach the goal.

In dynamic planning, the executing skeleton at the current time t needs to be checked for feasibility, both symbolically and geometrically. Formally, given the current inferred symbolic state s_t, the skeleton feasibility at the current time t is defined as:

$$\mathbf{F}(a_{1:K}, s_t) = \begin{cases} 1 & s_0 = s_t, \exists x : [t, KT] \to \mathcal{C} : (1b) - (1f) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

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, given the current symbolic and geometric state x_t, s_t. Initially, similar to single planning in Section II-A.3, the algorithm finds the best feasible skeleton initially and sets it to be the current executing skeleton $\kappa \in \Gamma$. For each replanning trigger, the algorithm checks for actual symbolic and geometric feasibility $\mathbf{F}(\kappa, s_t)$ of the current executing skeleton. If the skeleton κ is feasible, the NLP is optimized for κ from the elapsed time τ, 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 triggered to replan a new skeleton solution set $\Gamma_t(s_t, \mathbb{S}_{\text{goal}}, I)$ and optimize for the current best feasible skeleton. One may notice that the elapsed time τ 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

The motion prediction layer infers likely symbolic and geometric changes in the workspace, given an initial symbolic state s_t and a 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 feasibility of a skeleton κ can then be checked with $\mathbf{F}(\kappa, s_t)$ defined in Equation 4.

At the top level, our hierarchical motion prediction uses Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [15] and a low-level which performs full-body motion

Start State	(0, 4, 0, 1, 0, 3, 1, 0, 1, 2)
Actions	Go to white shelf Pick up cup Go to table Place
End State	(1, 3, 0, 1, 0, 3, 1, 0, 1, 0)

TABLE II: Example high-level trajectory

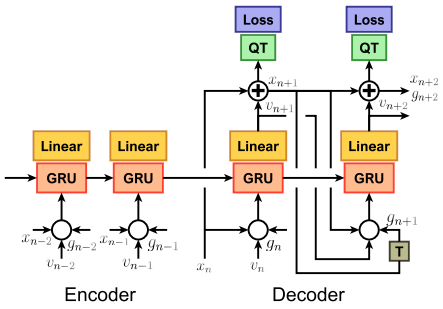


Fig. 1: Structure of the VRED [16] with the goal input added.

prediction conditioned on sequence of sub-goals induced by sequence of high-level actions given by the top-level.

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 [16] and make it goal-conditioned by adding a three-dimensional position g_t to the input of the network at every timestep (see Figure 1). The goal input g_t is relative to the coordinate frame of the human and thus changes every timestep. Particularly, we train two networks using human data from the MoGaze dataset, 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 loss between joint angles. We add an additional loss $l_{\text{goal}} = |g_T - \phi(x_T)|^2$ penalizing the distance between the input goal position and the predicted goal position at the last timestep T of the prediction, using a forward kinematic layer ϕ. After training, different trajectories can be generated by varying the goal input manually. Composing multiple subgoals can be used for sequential long-term motion predictions of the human. Note that we could also use other planning-based predictors together with MaxEnt IRL. However, we use VRED due to the scaling property of deep models that learns high-dimensional configuration trajectory of human-motion captures.

2) Network State Representation: We learn a policy π 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. MaxEnt IRL is based on state frequency calculations. For the MoGaze dataset (see Section III), the discretized state is given by the num-

```
(define (domain set_table)
  (:requirements :typing)
  (:types location object)
  (:constants
    table small_shelf big_shelf - location
    cup_red cup_green cup_blue cup_pink plate_pink plate_red plate_green
       plate_blue jug bowl - object
  )
  (:predicates
    (agent-at ?l - location)
    (on ?x - object ?l - location)
    (agent-free)
    (agent-avoid-human)
    (agent-carry ?x - object)
    (human-carry ?x - object)
  )

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

Fig. 2: PDDL-syntax symbolic domain of set-table task.

ber of objects on a location and the human position as follows: (cups-table, cups-shelf1, cups-shelf2, plates-table, plates-shelf1, jugbowl-table, jugbowl-shelf1, jugbowl-shelf2, humanPos). The action space is discretized similarly. An example skeleton can be seen in Table II.

We use heuristics for interfering with 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 [17]. The full long-term prediction is achieved by obtaining the skeleton from the high-level policy π, 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.

III. 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.

IV. Results & Conclusions

Here we report our evaluations and verify the effectiveness of the proposed framework.

A. Long-Term Motion Prediction using Hierarchies

In this experiment, the dataset is split into the training set and test set, in which the test set is the data recorded by a human participant not included in the training set. The report results will be in the test set. 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 angular loss of 7.99 instead of 10.14 and a significantly better position loss 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 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.

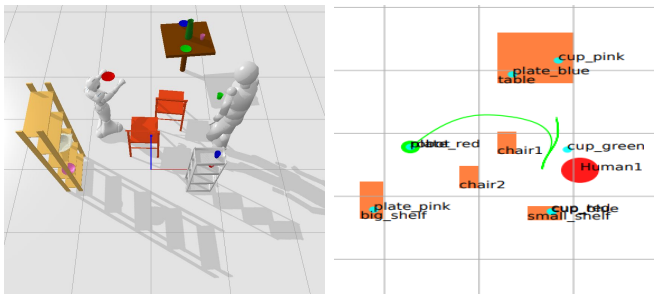
B. Dynamic LGP with Human Ground Truth

To test Dynamic LGP, we design the PDDL-syntax [18] domain following the available objects in the MoGaze dataset. Figure 2 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 = {(agent-free), (agent-avoid-human), (on cup-green big-shelf), (on plate-blue small-shelf)} and the goal state s_g = {(on cup-green table), (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 that 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. An example task instance is shown in Figure 3

We then run two planning modes; single planning and dynamic planning for each of the 63 segments. The task instance is considered successful if, at the end of the robot trajectory, the inferred symbolic state is in the goal set $\mathbb{S}_{goal}$. 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. Lower ratio means higher task time reduction.) and path ratio (i.e., the ratio of distance traveled by the robot,



(a) Initial states (b) Get blue plate

Fig. 3: (Left) Pepper carries a plate while the human is carrying a green cup to setup a dinner table. (Right) The motion plan resulting from Dynamic LGP based on current environmental conditions.

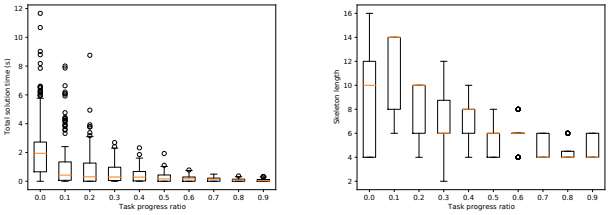


Fig. 4: Total time (left) to find an overall feasible solution and skeleton length (right) over task progress.

	Single planning	Dynamic planning
Success rate	84.1%	95.2%
Symbolic plan time	$0.032 \pm 0.036(sec)$	$0.045 \pm 0.053(sec)$
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 III: Dynamic LGP with Human Ground Truth

	Single planning	Dynamic planning
Success rate	91.2%	100%
Symbolic plan time	$0.0005 \pm 0.0001(sec)$	$0.0006 \pm 0.0002(sec)$
Task time reduction	0.298 ± 0.078	0.300 ± 0.100
Path ratio	1.000	0.626 ± 0.155
LGP replan count	-	3.0 ± 0.87

TABLE IV: Dynamic LGP with Human Prediction

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 the potential mismatch between the degraded ground truth and the actual ground truth. Moreover, we also see in Table III that the trajectory length of dynamic planning is almost half of single planning (i.e., path ratio), because the only human obstacle constraint is efficiently updated at every trigger.

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 inferred from the environment is rapidly changed.

Generally, the longer the action skeleton, the longer it takes to solve one NLP. The longest sequence length of 16 takes a median runtime of about 9 seconds, with ≈ 450 timesteps to optimize in an NLP. This timestep corresponds to the time discretization of interior-point trajectory optimization algorithm [13], [14]. Note that the longest skeleton only exists at the beginning of the task instance where the robot can wait to find an initial solution. As in Figure 4, as the task progresses, the total solution time to find a feasible NLP rapidly decreases to below 1 second since the action skeleton length also decreases. This benefits the real-time application, as the planning loop needs to be fast. The overall framework has reasonable performance to finish the task safely. Notice in Figure 4 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 described in Section II-B. We run 5 task instances for each segment to capture the human motion prediction statistics due to its stochasticity. The settings are the same as in Section IV-B. The overall task IoU between the robot and the human objects is 0.34 ± 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.

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 by a factor approaching 2, which is what one would expected when two agents collaborate at a task.

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, pp. 1930–1936.

[2] P. Kratzer, S. Bihlmaier, N. B. Midlagajni, R. Prakash, M. Toussaint, and J. Mainprice, "Mogaze: A dataset of full-body motions that includes workspace geometry and eye-gaze," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 367–373, 2020.

[3] A. M. Bauer, D. Wollherr, and M. Buss, "Human-Robot Collaboration - a Survey." *I. J. Humanoid Robotics*, vol. 05, no. 01, pp. 47–66, 2008. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219843608001303

[4] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, Nov. 2017. [Online]. Available: "http://dx.doi.org/10.1007/s10514-017-9677-2

[5] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2013.05.007

[6] G. Hoffman and C. Breazeal, "Cost-based anticipatory action selection for human–robot fluency," vol. 23, no. 5, pp. 952–961, 2007.

[7] R. Alami, A. Clodic, V. Montreuil, E. A. Sisbot, and R. Chatila, "Task planning for human-robot interaction," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, 2005, pp. 81–85.

[8] S. Lemaignan, M. Warnier, E. A. Sisbot, A. Clodic, and R. Alami, "Artificial cognition for social human–robot interaction: An implementation," *Artificial Intelligence*, vol. 247, pp. 45–69, 2017.

[9] M. Gharbi, R. Lallement, and R. Alami, "Combining symbolic and geometric planning to synthesize human-aware plans - toward more efficient combined search." *IROS*, pp. 6360–6365, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7354286/

[10] B. Busch, M. Toussaint, and M. L. 0001, "Planning Ergonomic Sequences of Actions in Human-Robot Interaction." *ICRA*, 2018. [Online]. Available: https://dblp.org/rec/conf/icra/BuschT018

[11] M. Toussaint and M. Lopes, "Multi-bound tree search for logic-geometric programming in cooperative manipulation domains," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4044–4051.

[12] M. Fox and D. Long, "Pddl2.1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, p. 61–124, Dec 2003. [Online]. Available: http://dx.doi.org/10.1613/jair.1129

[13] J. Mainprice, N. Ratliff, M. Toussaint, and S. Schaal, "An interior point method solving motion planning problems with narrow passages," in *IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, 2020.

[14] J. Mainprice, "bewego: Differentiable kinetics optimization library," 2019–2021. [Online]. Available: https://github.com/humans-to-robots-motion/bewego

[15] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8, 2008, pp. 1433–1438.

[16] H. Wang and J. Feng, "Vred: A position-velocity recurrent encoder-decoder for human motion prediction," *arXiv preprint arXiv:1906.06514*, 2019.

[17] P. Kratzer, N. B. Midlagajni, M. Toussaint, and J. Mainprice, "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.

[18] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins, "Pddl-the planning domain definition language," 1998.