Workspace Optimization Techniques to Improve Prediction of Human Motion During Human-Robot Collaboration

Yi-Shiuan Tung
yi-shiuan.tung@colorado.edu
University of Colorado Boulder
Boulder, USA

Matthew B. Luebbers
matthew.luebbers@colorado.edu
University of Colorado Boulder
Boulder, USA

Alessandro Roncone
alessandro.roncone@colorado.edu
University of Colorado Boulder, Lab0 Inc.
Boulder, USA

Bradley Hayes
bradley.hayes@colorado.edu
University of Colorado Boulder
Boulder, USA

ABSTRACT

Understanding human intentions is critical for safe and effective human-robot collaboration. While state of the art methods for human goal prediction utilize learned models to account for the uncertainty of human motion data, that data is inherently stochastic and high variance, hindering those models' utility for interactions requiring coordination, including safety-critical or close-proximity tasks. Our key insight is that robot teammates can deliberately configure shared workspaces prior to interaction in order to reduce the variance in human motion, realizing classifier-agnostic improvements in goal prediction. In this work, we present an algorithmic approach for a robot to arrange physical objects and project "virtual obstacles" using augmented reality in shared human-robot workspaces, optimizing for human legibility over a given set of tasks. We compare our approach against other workspace arrangement strategies using two human-subjects studies, one in a virtual 2D navigation domain and the other in a live tabletop manipulation domain involving a robotic manipulator arm. We evaluate the accuracy of human motion prediction models learned from each condition, demonstrating that our workspace optimization technique with virtual obstacles leads to higher robot prediction accuracy using less training data.

CCS CONCEPTS

• **Computing methodologies → Robotic planning**; **Planning under uncertainty**; • **Human-centered computing → Mixed / augmented reality**.

KEYWORDS

motion prediction, human-robot collaboration, environment adaptation, augmented reality, legibility

ACM Reference Format:
Yi-Shiuan Tung, Matthew B. Luebbers, Alessandro Roncone, and Bradley Hayes. 2024. Workspace Optimization Techniques to Improve Prediction of Human Motion During Human-Robot Collaboration. In *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3610977.3635003

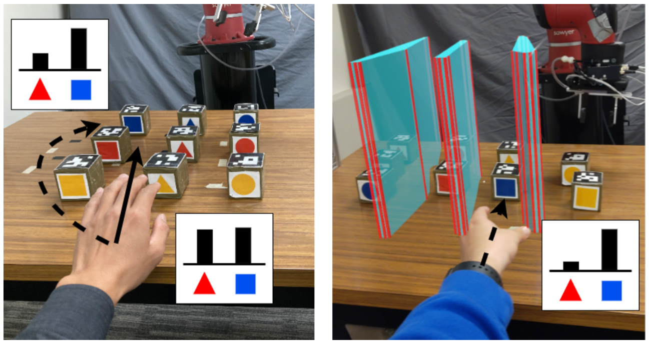


Figure 1: Workspace configuration affects the robot's ability to correctly predict the human's goal – the blue square cube. Left: The legible path (dotted) requires the human to take a circuitous route while the natural path (solid) is not legible. Right: Our approach generates a workspace configuration by arranging physical objects and projecting "virtual obstacles" in AR (cyan and red barriers), in order to induce naturally legible paths from the human.

1 INTRODUCTION

In human-robot collaborative tasks, shared mental models between agents enable the awareness and joint understanding required for effective teamwork [32]. With no shared notion of the task to be completed, the inherent stochasticity and opacity of human decision-making makes robot planning difficult [30]. To this end, prior research efforts have focused on developing robots that can predict human behavior [15, 19, 29], generating motion plans to safely interact in a shared environment [11, 26]. However, these methods are limited by the quality of robot predictions of a human collaborator's intention and resultant behavior. With inaccurate human models or unexpected human behavior diverging from past

HRI '24, March 11–14, 2024, Boulder, CO, USA
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0322-5/24/03…$15.00
https://doi.org/10.1145/3610977.3635003

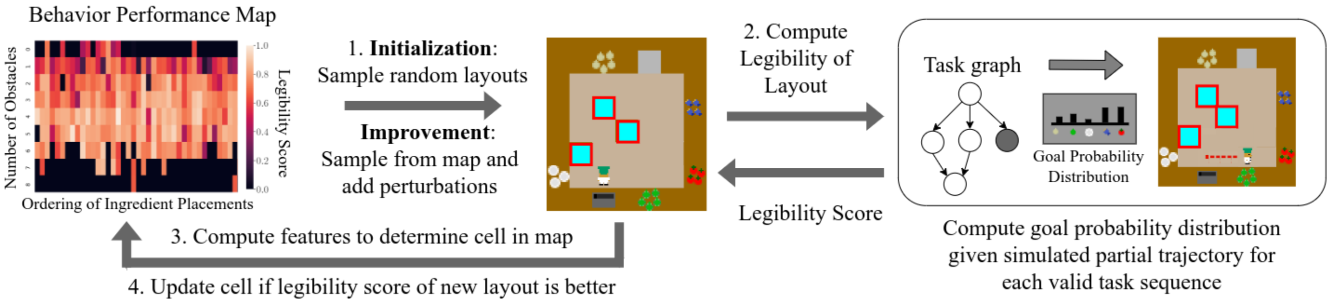


Figure 2: Our approach for generating workspace configurations that enable accurate human goal predictions. (1) In the initialization phase, we sample random environment layouts to populate the behavior performance map, which stores diverse and high performing solutions. This is followed by the improvement phase where we sample directly from the map and add perturbations to test whether the legibility is improved. (2) In both phases, we compute the legibility of the sampled layout by computing the probability of predicting the correct goal at each stage of the task execution. (3) We compute the features of the sampled layout to determine its location in the map. (4) The map is updated if the legibility score of the sampled layout is better than the existing one.

experiences, the robot may produce unsafe interactions [20], especially in safety-critical or close-proximity settings [36].

To address the inherent challenges of accurately predicting human motion early in a demonstrated trajectory, our key insight is that robots can take an active role in structuring the environment to reduce the variance of human motion caused by dense and overlapping task spaces, thereby improving the performance of human behavior models. In this work (see Fig. 1), we introduce an algorithmic approach for a robot to configure a shared human-robot workspace prior to interaction in order to improve a robot's ability to predict the human collaborator's goals during task execution. As detailed in Fig. 2, we present an objective function that scores potential workspace configurations in terms of how legible the actions of a human teammate are likely to be when performing a task in that environment. We use the mathematical formulation of legibility from [5], which computes the probability of successfully predicting an agent's goal given an observation of a snippet of its trajectory. Our approach finds workspace configurations that maximize legibility over the valid goals at each stage of task execution.

Each candidate workspace configuration combines a potential arrangement of physical objects and projection of "virtual obstacles" in augmented reality (AR) in the environment. While the arrangement of physical objects can be achieved prior to the shared task via a simple composition of robotic pick and place actions, the addition of augmented reality-based virtual obstacles is particularly effective in imposing explicit constraints on the possible motions of the human, without requiring additional physical changes to the environment. Note that our approach does not restrict the human to a particular goal and retains their ability to decide on the goal they're reaching towards. Humans also have the flexibility to alter their decisions midway through a task, since goal prediction is performed at every time step.

We efficiently explore the space of workspace configurations using a quality diversity (QD) algorithm called Multi-dimensional Archive of Phenotypic Elites, or MAP-Elites [22]. Instead of finding a single optimal solution, MAP-Elites produces a map of performant solutions along dimensions of a feature space chosen by the designer. MAP-Elites enables efficient and extensive exploration of complex search spaces, leading to higher quality solutions as compared to other search algorithms [4, 24].

We empirically demonstrate that our workspace optimization approach improves the accuracy of human goal prediction models, showing results for both a Bayesian predictor with learned cost functions using Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) and a time series multivariate Gaussian model. We collect human motion data in an online 2D navigation experiment and a real world tabletop experiment involving a robotic manipulator, comparing our approach to other workspace arrangement strategies. In summary, we present two primary contributions: 1) an algorithm for optimizing the placement of physical and virtual objects in a shared human-robot workspace for maximizing human legibility, and 2) an evaluation of that algorithm via two human subjects experiments, showing that it influences human behavior in ways that improve a robot collaborator's prediction model.

2 RELATED WORK

Planning using Human Motion Models. In human-robot collaboration, the robot needs to predict human motion in order to coordinate its actions with those of the human. Prior work has developed motion planning algorithms that models human motion and generates robot plans to safely interact with humans in tabletop [14] and navigation [10, 27] settings. To account for the uncertainty in human motion prediction, prior work has used partially observable Markov decision processes (POMDPs) to determine optimal actions for a robot given a probabilistic belief over the human's intended goals [11, 26]. These methods rely on the human motion model to achieve safe interactions, but human motion is inherently highly variable as humans can always move unexpectedly. The difficulty of the problem can be seen by the variety of approaches taken toward accurate human motion prediction: Gaussian models [17, 28], dynamic movement primitives [18, 37], latent representation learning [2, 21], and imitation learning [6, 19], among various other methods [29]. Our work takes a different approach and addresses a fundamental challenge faced by all human motion prediction models; we reduce the uncertainty inherent in modeling the intentions of human collaborators by pushing them towards legible behavior via environment design. Our work improves human motion model predictions by increasing environmental structure to reduce uncertainties, facilitating more fluent human-robot interactions.

Environment Design in Robotics: Prior work has also explored designing or modifying environments in order to better achieve agent goals [33, 34]. Zhang et al. [38] proposed a framework for designing environments that optimize an agent's reward, and Keren et al. [12] extended it for stochastic transitions. Kulkarni et al. [13] generate interpretable robot behaviors by modifying the environment. These works show the potential advantages of a robot using environment design to improve its task performance and interpretability. There has also been work on modifying the environment for collaborative teaming. Zhang et al. [39] optimizes warehouse layouts for multi-robot coordination, and Bansal et al. [1] explores the idea of robots moving objects to reduce the likelihood of future collisions in a tabletop task. Our technique differs from prior work in that the robot modifies its environment with the explicit goal of improving its ability to predict human behavior for fluent collaboration with human teammates.

Finding the optimal environment by simply iterating through all possible workspace configurations quickly becomes intractable as the number of objects or possible states increases. To address this, others have used quality diversity (QD) algorithms to generate diverse environments for evaluating the performance of shared autonomy algorithms [9] and explore diverse coordination behaviors as a result of environment design [8]. Inspired by the success of QD algorithms in finding diverse solutions in large search spaces [4, 24], we use MAP-Elites to search for environments that best elicit legible human behavior. Our work extends QD approaches to generate interaction scenarios that influence human behavior and address the robot's limitations when collaborating with humans.

3 LEGIBLE WORKSPACE GENERATION

In this section, we describe our approach for modifying the shared human-robot workspace to maximize legibility and enable more accurate human goal predictions (summarized in Fig. 2).

3.1 Legibility Score

To evaluate the legibility of a workspace configuration, we consider the probability distribution of predicting that the human is approaching goal G given an observed trajectory from start state S to intermediate point Q. We use the formulation developed by [5] shown in Equation 1.

$$\Pr(G|\xi_{S\to Q}) \propto \frac{exp(-C(\xi_{S\to Q}) - C(\xi^*_{Q\to G}))}{exp(-C(\xi^*_{S\to G}))} \tag{1}$$

The optimal human trajectory from point X to point Y with respect to cost function C is denoted by $\xi^*_{X\to Y}$. Equation 1 evaluates how cost efficient (with respect to C) going to goal G is from start state S given the observed partial trajectory $\xi_{S\to Q}$ relative to the most efficient trajectory $\xi^*_{S\to G}$.

Let $\mathcal{G}$ be the set of valid goals at the current time step. We develop a legibility score (Eqn. 2) for use in our optimization objective that, for every valid goal at a given time step in the task execution, maximizes the margin of prediction between the human's chosen goal $G_{true} \in \mathcal{G}$ and all other valid goals. If the predicted goal is not G_{true}, the score is penalized by a fixed cost c multiplied by the length of the sampled human trajectory $|\xi_{S\to Q}|$. Otherwise, the score is the difference of the two highest probabilities (computed by the *margin* function). The notation $G_{(i)}$ denotes the i-th index of a sorted list of length n that represents the goal probabilities ordered from least to most likely given the observed trajectory $\xi_{S\to Q}$.

$$\text{EnvLegibility}(G_{true}) = \begin{cases} -c|\xi_{S\to Q}|, & \text{if } \arg\max_{G\in\mathcal{G}} \Pr(G|\xi_{S\to Q}) \neq G_{true} \\ margin(\mathcal{G}|\xi_{S\to Q}) = G_{(n)} - G_{(n-1)}, & \text{otherwise} \end{cases} \tag{2}$$

3.2 Optimization for Task Legibility

To generate a workspace configuration with improved legibility of the agent's goals for a task, we maximize the legibility score from Equation 2 for all valid subtask sequences (Eqn. 3). We use precedence constraints introduced by the structure of the task (i.e., which subtasks are prerequisites for other subtasks) to identify the set of valid subtasks (and thus valid goals $\mathcal{G}$) at any given time step.

$$\max \sum_{T'\in\text{permutations}(T)} \mathbb{1}\{\text{valid}(T')\} \times \sum_{t\in T'} \sum_{G\in\mathcal{G}} \text{EnvLegibility}(G) \tag{3}$$

Let T' represent a valid subtask sequence, consisting of all k subtasks of task T: $t_1, ..., t_k$, ordered such that for each index i, subtask t_i has all precedence constraints satisfied. Each subtask has one or more goals that an agent can reach to complete it. For example, the task "Set Table" may have a subtask "Get/Place plate on place mat" with multiple satisfying goals (multiple place mats). Additionally, since there are often multiple valid subtask sequences given an observed set of subtasks $t_1, ..., t_i$, $\mathcal{G}$ represents the set of goals corresponding to all uncompleted subtasks with satisfied precedence constraints. For example, after a plate is set, "Get/Place Fork" and "Get/Place Spoon" may be equally valid as the next subtask to perform. At this point, $\mathcal{G}$ includes potential goals for both fork and spoon placement subtasks. The objective function considers all possible goals that the human might be reaching for at a given stage of task execution and maximizes the probability of correctly predicting the human's chosen goal.

3.3 Search using Quality Diversity

Iterating through all possible workspace configurations to find the optimal solution is intractable since the number of possible configurations is exponential in the number of goals, virtual obstacles, and size of the workspace. We use MAP-Elites [22] to approximate the optimal solution.

In Algorithm 2, MAP-Elites maintains a behavior performance map, or solution map S, that stores high performing solutions across features or behaviors of interest. To find the most legible workspace (objective function F), the designer chooses a set of features or behaviors (computed by measure function M) such as the distance between the objects or the number of virtual obstacles. The algorithm would then find the most legible workspace for each possible combination of features found. As input, the algorithm also requires a model G_H that outputs human trajectory given a goal. G_H can be learned from data via inverse optimal control [19] or approximated via shortest path to goal [5].

Algorithm 1: improve_workspace

```
   Input: Workspace configuration $w$, objective score of w $s_w$,
          objective function $F$, measure function $M$, Solution
          map S, Solution values V
   Initialize: Best configuration $w^* \leftarrow w$, Best score $s^* = s_w$
1  $A$ = sample_perturbations($w$)
2  for $a \in A$ do
3      Generate new workspace $w'$ = apply_perturbation($w$, $a$)
4      Compute legibility score $s_{w'} = F(w')$
5      if $s_{w'} > s^*$ then
6          Update best workspace $w^* = w'$
7          Update best score $s^* = s_{w'}$
8      Determine features $\boldsymbol{m} = M(w')$
9      if $S[\boldsymbol{m}] = \emptyset$ or $s_{w'} > V[\boldsymbol{m}]$ then
10         Update solution map $S[\boldsymbol{m}] = w'$
11         Update solution values $V[\boldsymbol{m}] = s_{w'}$
12 if $w^* \neq w$ then
13     return improve_workspace($w^*$)
14 return $w^*$
```

Algorithm 2: Workspace Generation with MAP-Elites

```
   Input: Human Trajectory Generator $G_H$, Objective
          function $F$, measure function $M$
   Initialize: Solution map $S \leftarrow \emptyset$, Solution values $V \leftarrow \emptyset$
1  for $i = 1, ..., N$ do
2      if $i < N_{init}$ then
3          Generate workspace $w$ = random_workspace()
4      else
5          Sample workspace from map $w$ = random(S)
6          Run $w$ = improve_workspace($w$)
7      Determine features $\boldsymbol{m} = M(w)$
8      Determine objective score $s = F(w)$
9      if $S[\boldsymbol{m}] = \emptyset$ or $s > V[\boldsymbol{m}]$ then
10         Update solution map $S[\boldsymbol{m}] = w$
11         Update solution values $V[\boldsymbol{m}] = s$
12 return S, V
```

MAP-Elites (Alg. 2) consists of two phases: initialization and improvement. In the initialization phase, we randomly sample workspaces for N_{init} iterations (Line 3) and store them in the corresponding cell in the solution map by computing the features (Lines 7-11). In the solution map S, the cell associated with the vector of feature values m is denoted $S[m]$. For $N - N_{init}$ iterations, we perform the improvement phase: we first randomly sample from the solution map (Line 5) and then empirically approximate the gradient (Line 6) to improve the solution. Following differentiable QD [7], we use gradient information to speed up search. Unlike [7], our implementation only estimates gradients of the objective function and not the measure function. Since we empirically approximate the gradient using stochastic sampling (Alg. 1), the environment updates may not always align with the steepest ascent direction. This can lead to "suboptimal" solutions that can, however, contribute to increased diversity.

As detailed in Algorithm 1 (the *improve_workspace* function), Line 1 samples perturbations to the workspace (i.e. changing an item's position, adding or removing a virtual obstacle). For each perturbation, a new workspace configuration w' is generated by applying the perturbation. We keep track of the current best workspace w^* in terms of the objective score (Lines 5-7) and update the solution map if a better environment was found for the features $\boldsymbol{m}$ (Lines 9-11). If there was an improvement to the workspace, we run Algorithm 1 again (Lines 12-13). Otherwise, a local minima has been found, and we return the best workspace found (Line 14).

4 EVALUATION

We evaluate our approach in two environments: a) single player Overcooked [3] and b) tabletop collaborative pick and place with a Sawyer robot. These experiments validate our approach across navigation and tabletop domains while analyzing two distinct forms of human-borne motion data. To predict the human's goal given a partial trajectory in Overcooked, we use the Bayesian predictor developed by Dragan et al. [5] (Eqn. 1). For the tabletop experiment, we implement the time series multivariate Gaussian model proposed in [28] to predict which cube the human is reaching for. We chose these models as representative examples of two distinct approaches to human motion prediction. Collectively, these experiments demonstrate the broad applicability of our approach across varied environments and tasks, as well as varied human goal classification techniques.

4.1 Hypotheses

H1: Environments generated by our approach will enable more accurate predictions of the human goal throughout the task execution compared to baseline environments.
H2: The prediction models will have better accuracy than predictions based on heuristics such as predicting the goal that is closest to the current trajectory.
H3: Prediction models trained in environments generated by our approach will be more data-efficient, requiring fewer examples to reach peak performance levels as compared to those trained in baseline environments.

4.2 Overcooked Experiment

4.2.1 Experimental Setup. The Overcooked game (Fig. 3a-d) requires participants to fetch ingredients, place them in a pot, plate the cooked dish, and deliver the dish to a serving station in a grid world environment. We chose to run the experiment with a single human agent in order to isolate the effects of environment design on the goal predictability of human agents. An investigation on combining robot policy and environment design to influence goal prediction is an interesting direction for future work. We conducted an IRB-approved Amazon Mechanical Turk experiment with 20 participants aged 18 or older and with at least a 95% approval rating. Each participant played in five rounds—one training round followed by four conditions in randomized order. Each round had three soup deliveries with the following ingredients: 2 tomatoes + 1 onion, 2 onions + 1 cabbage, and 3 fish. The soups and the ingredients could be delivered or placed in the pot in any order. At each time step, we recorded the game state and action taken by the participant.

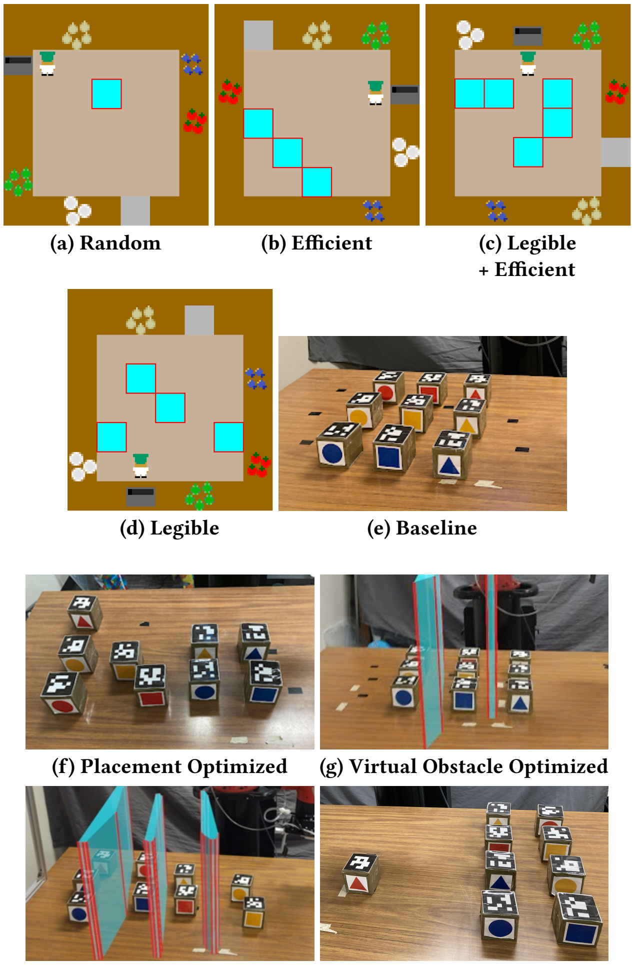
4.2.2 Goal Prediction Model. We use the Bayesian formulation (Equation 1) to predict the human goal in the Overcooked experiment. The predicted goal is given by $arg\max_G P(G|\xi)$ where ξ is the observed partial trajectory. We use Maximum Entropy Inverse Reinforcement Learning (MaxEntIRL) [40] to estimate the cost function C_θ that is modeled as a linear function of the state features f_s, $C_\theta(s) = \theta^T f_s$. We used the position of the player as the features in our experiments. The cost of a trajectory is the sum of the cost of states in the trajectory. MaxEntIRL models the trajectories ξ from the training data as a Boltzmann distribution $p_\theta(\xi) = \frac{1}{Z} exp(-C_\theta(\xi))$, where Z is the partition function computed via dynamic programming. We optimize the cost function parameters θ by maximizing the likelihood given training data $\max_\theta \sum_{\xi \in D} p(\xi|\theta)$ [40].

4.2.3 Conditions. We compare four conditions (Fig. 3a-d):
Random: We sample 1000 environments where each workstation is randomly placed in the perimeter, and each inner cell has a 20% chance of being assigned a virtual obstacle. This is the same as the initialization step of the MAP-Elites implementation. We then sort the layouts by the legibility score and select the layout with the median score (Fig. 3a).
Efficient: We use MAP-Elites to minimize the distance traveled to complete a task (Fig. 3b). This condition represents a common objective when designing workspace configurations – optimizing for task efficiency.
Legible-Efficient: The MAP-Elites objective function includes terms for both maximizing legibility and minimizing the distance traveled (Fig. 3c).
Legible: Our approach – using MAP-Elites to maximize legibility (Fig. 3d).

4.2.4 Workspace Generation. The workstations of the Overcooked game consist of the cooking pot, dish, serving, and four ingredient stations: onions, tomatoes, cabbages, and fish. For multiple agents to efficiently collaborate, an agent has to predict the ingredients the other agents are picking up so that efforts aren't duplicated. Therefore, for the objective function and evaluation, we only consider trajectories starting from the pot or serving stations and ending at the dish or ingredient stations. We allow the algorithm to move the workstations in the perimeter which is representative of reorganizing objects. The algorithm can also add or remove virtual obstacles onto any space within the perimeter. We add a constraint that the workstations have to be at least two cells away so that trajectories will be longer than a single step.

All conditions, except the Random condition, are generated using MAP-Elites with different objective functions. The features are the number of obstacles and the ordering of workstation placements from top to bottom and left to right. Figure 4 shows the behavior performance map generated by MAP-Elites. After 1000 iterations of the improvement phase, we use the solution with the highest legibility score in the map.

1) Initialization. The workspace is randomly sampled by finding non-overlapping placements for each workstation in the perimeter. For each cell not in the perimeter, there is a 20% chance the cell is a virtual obstacle. We verify that the sampled workspace has valid paths between the workstations before placing it in the behavior performance map. This phase is run for 1000 iterations.



(a) Random (b) Efficient (c) Legible + Efficient

(d) Legible (e) Baseline

(f) Placement Optimized (g) Virtual Obstacle Optimized

(h) Both Optimized (i) Desired configuration

Figure 3: (a-d) Overcooked layouts and (e-h) initial cube configurations used in our experiments. The environments generated by our approach, (d) *Legible* and (h) *Both Optimized*, optimize object and virtual obstacle (shown in cyan with red edges) placements to elicit legible human motion. (i) The tabletop experiment involves the human and the robot collaboratively placing cubes into the desired configuration – two columns on the right with a given ordering.

2) Improvement. The available perturbations are moving a station to an empty counter, swapping positions with another station, and adding or removing virtual obstacles.

4.3 Tabletop Experiment

4.3.1 Experimental Setup. We conducted an IRB-approved human subjects study with 12 participants (8 male, 4 female), with an age range of 19 to 31 years old (M = 24.42, SD = 3.37), recruited from a university campus to collect human hand trajectory data in different tabletop workspace setups. We collected 8 trajectories per participant for each of the four conditions, presented in a randomized, counterbalanced order. The study involves the human and an autonomously operating Sawyer 7 DoF manipulator robot working

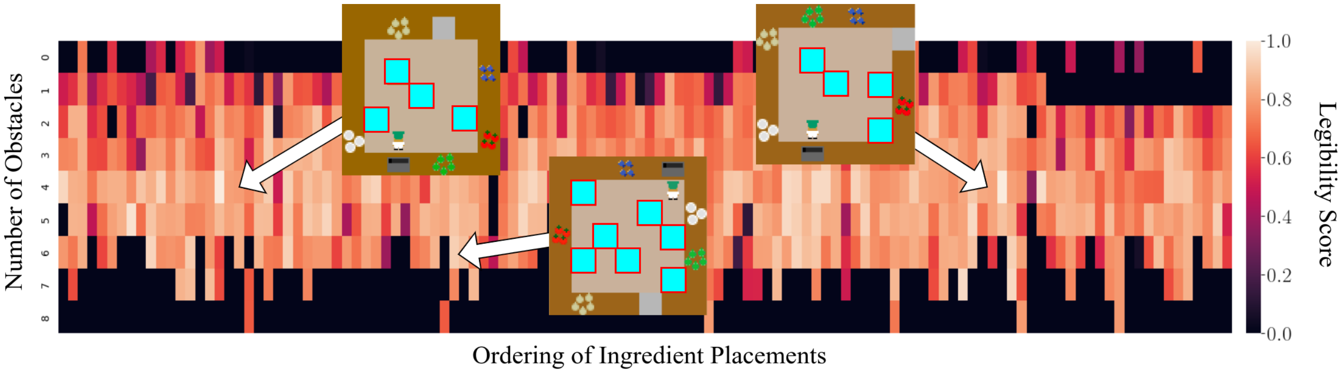
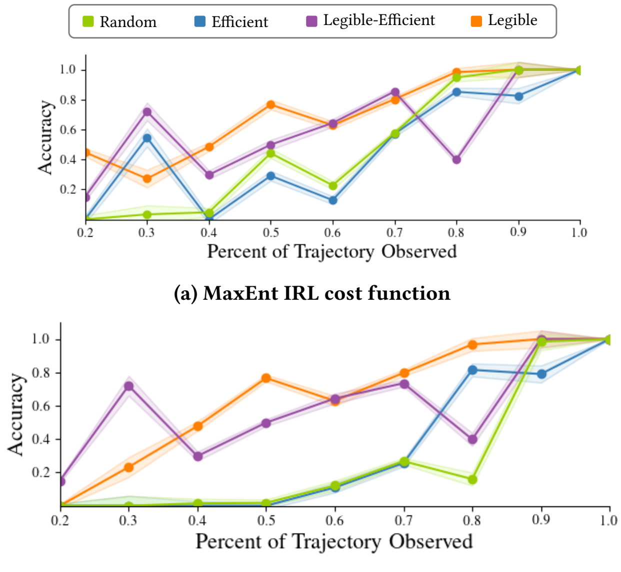


Figure 4: Behavior performance map generated in the Overcooked domain. MAP-Elites is able to explore legible workspace configurations across combinations of the features: number of obstacles and the ordering of ingredient placements, which improves the best solution found in complex search spaces.

collaboratively to place cubes into a desired configuration (Fig. 3i). To prevent the robot from picking up the same cube, the participant is asked to first reach for a cube while the robot maintains a probability distribution over the possible cubes the human is reaching for in real time. Once the robot is sufficiently confident of the human's goal, the robot will select its own cube to pick up and move to grasp it. The human and robot team continues to pick a cube each until the task is completed. The precedence constraints are set such that the first column of the desired configuration must be completed before the second column can start. This experiment evaluates homogeneous teams where the human and robot are able to perform all subtasks. Our approach generalizes to heterogeneous teams, in which case we modify T from Equation 3 to only include subtasks the human can perform.

4.3.2 *Goal Prediction Model.* We use a time series multivariate Gaussian model [28] to predict human goals given the human hand trajectory collected from the tabletop experiment. For a training data set D, we use dynamic time warping (DTW) [23] to align the trajectories so they all have length K. For each time step $k \in K$ and goal G, we train a multivariate Gaussian with mean $\mu_G[k] = \frac{1}{|D|}\sum_{i=1}^{|D|} f_i[k]$ and covariance $\Sigma_G[k] = \frac{1}{|D|-1}\sum_{i=1}^{|D|}(f_i[k]-\mu[k])(f_i[k]-\mu[k])^T$ where $f_i[k]$ are the feature values of the ith trajectory at time step k. At prediction time, the probability of a goal G given a partial trajectory ξ is $P(G|\xi) \propto \prod_{k=1}^{K}[\mathcal{N}(\mu_G[k], \Sigma_G[k])]^{\frac{1}{K}}$ [28]. The goal with the highest probability is the predicted goal. We use two features, the x and y positions of the hand, captured via an Intel RealSense RGB camera and a real-time hand tracking algorithm [35].

4.3.3 *Conditions.* We compare workspace setups using our optimization against **Baseline**, where the cubes are initially sorted by their color and shape (Fig. 3e). We also perform an ablation study, removing the algorithm's ability to organize the cubes or project virtual obstacles. As such, the remaining conditions are:
Placement Optimized: We optimize for legibility by only rearranging the cubes, with no virtual obstacles (Fig. 3f).
Virtual Obstacle Optimized: The cubes are sorted by their color and shape, and we optimize for legibility by projecting virtual obstacles (Fig. 3g).



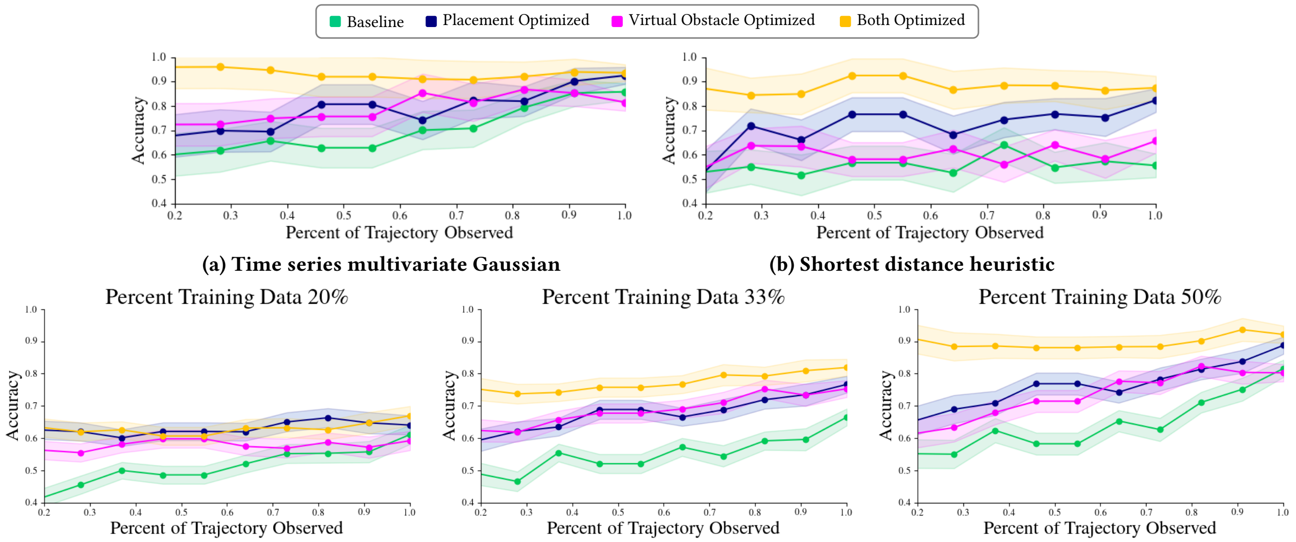
(a) MaxEnt IRL cost function

(b) Shortest distance heuristic

Figure 5: Overcooked experiment results: (a) Mean accuracy of Bayesian predictor using cost function learned from MaxEnt IRL. Compared to the shortest distance heuristic shown in (b), MaxEnt IRL improved accuracy the most when less than 60% of the trajectory has been observed. The width of the shaded area is the Tukey's Q critical value, so that overlaps indicate no statistical significance. The *Legible* condition elicits significantly higher accuracy for most cases.

Both Optimized: Our approach that optimizes for legibility by rearranging the cubes and projecting virtual obstacles (Fig. 3h).

4.3.4 *Workspace Generation.* The feature dimensions for MAP-Elites are the minimum distance between the cubes and the ordering of cubes by the x-axis. In contrast to the layout generation for the Overcooked game, we do not add virtual obstacles in the initialization step. Due to the continuous state space and the difficulty of randomly sampling a useful virtual obstacle, we choose fixed size virtual obstacles and insert them between two randomly selected



(a) Time series multivariate Gaussian

(b) Shortest distance heuristic

(c) Time series multivariate Gaussian learning curve

Figure 6: Tabletop experiment results: (a) Mean accuracy of time series multivariate Gaussian model. Our approach, *Both Optimized*, elicits significantly higher or comparable accuracy compared to the baselines. (b) Mean accuracy using shortest distance heuristic where the closest cube to the current hand position is the predicted goal. (c) Mean accuracy of time series multivariate Gaussian model with varying amounts of training data. *Both Optimized* elicits significantly higher accuracy compared to baselines starting at 33% training data, achieving around 90% accuracy with just 50% of the available training data.

cubes in the improvement phase. By employing this heuristic, we can effectively explore configurations that result in altered human reaching motions, which are approximated using shortest paths in a visibility graph. We experimented with virtual obstacles of different sizes and found that obstacles measuring 30cm x 1cm were sufficient in inducing distinct reaching motions.

1) Initialization. We randomly sample (x, y) positions for the cubes and ensure that they don't overlap. The positions are continuous values uniformly sampled within the permissible boundaries.

2) Improvement. The improvement step consists of first changing cube positions and then adding virtual obstacles. We sample new cube positions from a Gaussian centered at the cube with variance = 7cm. The virtual obstacles are placed between two randomly sampled cubes.

The boundaries of the algorithmically generated virtual obstacles are passed to an AR interface, implemented using a Microsoft HoloLens 2 head-mounted display. The AR interface renders those obstacles directly in the environmental context of the shared workspace as holograms of cyan barriers with red outlines (Fig. 3h). These barriers appear to the human as if they are physically located in the environment, and indicate regions of the environment the human should not enter.

5 RESULTS

In both experiments, we run a stratified 4-fold cross validation with 3 repeats and different randomization in each repetition. The stratified cross validation preserves the percentage of samples for each class (i.e. the valid goals at each time step) in each fold. Figures 5 and 6 show the mean goal prediction accuracy as a function of the percentage of total trajectory observed for Overcooked and tabletop experiments respectively. A prediction is defined as correct if the single highest probability value in the probability distribution is the same as the target goal. A trajectory is the human motion when reaching towards a goal: x, y positions in the Overcooked grid and hand positions in the x-y plane for the tabletop experiment. We use the one-way analysis of variances (ANOVA) and perform post-hoc analysis using Tukey's HSD test for multiple comparisons to test for effects between condition pairs. We plot Tukey's Q critical value as the width of the shaded area in Figures 5 and 6 such that overlaps indicate an absence of a statistically significant difference.

5.1 Overcooked

Figure 5a shows the prediction results using the Bayesian predictor. The *Legible* condition elicits significantly higher or comparable accuracy than the best performer among other conditions except when 30% of the trajectory has been observed, supporting hypothesis *H1*. The counterintuitive bump at 30% is because the optimization penalizes incorrect predictions more for longer trajectories (Eqn. 2). Thus, it made a trade-off: the accuracy is lower when predicting goals about 30% into a subtask but higher for longer trajectories. Figure 5b shows the same analysis but we use the shortest Manhattan distance as a heuristic for the cost function. The mean accuracy is lower when only a small percentage of the trajectory has been observed compared to the predictions with learned cost functions, suggesting that IRL enables faster accurate goal predictions which supports hypothesis *H2*.

5.2 Tabletop Experiment

Figure 6a shows the mean goal prediction accuracy when different percentages of the total trajectory is observed using the Gaussian model. The environment generated by our approach, *Both Optimized*, elicits significantly higher prediction accuracy than baseline

Table 1: The average determinant (det) and trace of the covariance matrices of the multivariate Gaussian models.

	Baseline	Placement Optimized	Virtual Obstacle Optimized	Both Optimized
Det	9.36e-06	8.50e-06	6.14e-06	**1.09e-06**
Trace	7.12e-03	5.90e-03	3.90e-03	**2.77e-03**

environments when less than 50% of the trajectory has been observed. When using the shortest distance heuristic (Fig. 6b), where the closest cube to the current hand position is the predicted goal, all conditions elicit lower accuracy compared to the predictions when using the Gaussian models. The shortest distance heuristic exhibits poor performance even when the entire trajectory is observed. Both hypotheses *H1* and *H2* are supported by these results.

To evaluate the learning curve of the Gaussian model, we perform stratified splits cross validation with 10 repeats. Figure 6c shows the performance as training data increases. *Both Optimized* elicits significantly higher accuracy than the baselines with 33% training data and achieves around 90% accuracy with 50% training data, supporting our hypothesis *H3* that our approach generates environments where the training of prediction models is more data-efficient.

In Figure 7, we plot the mean and covariance of the learned time series multivariate Gaussian for each condition. Qualitatively, the Gaussian models in the *Both Optimized* condition have less covariance, a measure of uncertainty, compared to the Gaussian models trained in baseline environment configurations. Table 1 shows the average determinant and trace of the covariance matrices of the Gaussian models. The determinant is a measure of the magnitude of variation among the variables, and the trace is the sum of the variances of the individual variables but does not consider the correlations between the variables. The Gaussian models trained in the *Both Optimized* layout have lower values for both the determinant and trace compared to the models trained in baseline environments.

Overall, our results show that our workspace configuration approach reduces the uncertainty in human motion data and improves the accuracy of goal prediction models in a planar navigation and tabletop manipulation tasks. The models are also more data-efficient in our generated environments, requiring less data to reach their best performance levels as compared to those trained in baseline environments.

6 CONCLUSION

In this work, we introduce an algorithmic approach for autonomous workspace optimization to improve robot predictions of a human collaborator's goals. By rearranging physical objects in the workspace and projecting AR-based virtual obstacles into the environment prior to interaction, the robot influences the human into more legible behavior during task execution, thereby reducing the uncertainty of their motion. Through dual experiments in 2D navigation and tabletop manipulation, we show that our approach results in more accurate model predictions across two distinct goal inference methods, requiring less data to achieve these correct predictions. Importantly, we demonstrate that environmental adaptations can be discovered and leveraged to compensate for shortfalls of prediction models in otherwise unstructured settings.

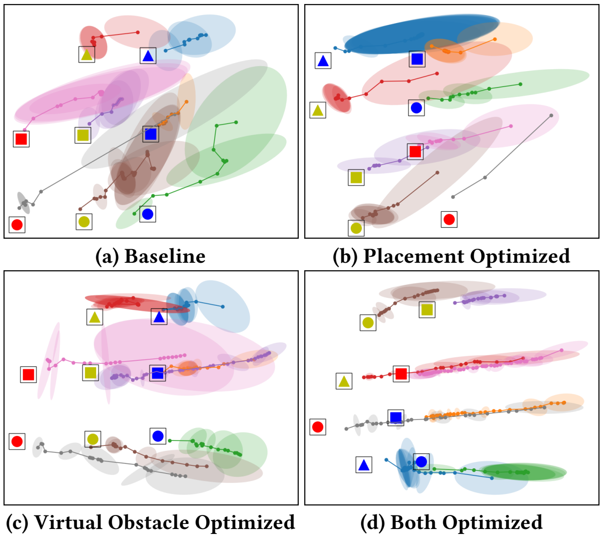

(a) Baseline
(b) Placement Optimized
(c) Virtual Obstacle Optimized
(d) Both Optimized

Figure 7: Top down view of the workspace plotting the mean and covariance of the time series multivariate Gaussian for each condition. The Gaussian models are fit to the hand trajectory data and predicts the human's goal (section 4.3.2). The model in the *Both Optimized* condition has less covariance compared to the models trained in the other environment configurations.

Our approach is applicable for domains where the following conditions hold: 1) multiple agents share the same physical space and the agents do not have access to other agents' controllers or decision making processes, 2) the environment allows physical or virtual configurations, and 3) environment configuration can be performed prior to the interaction. By improving goal prediction, we envision that our framework enhances human-robot teaming across domains such as shared autonomy for assistive manipulation, warehouse stocking, and cooking assistance, among others. Through our results, we demonstrate the generalizability of our overall approach across varied tasks, environments, and human motion prediction models, showing its potential to realize fluency improvements in a number of human-robot collaborative domains.

One limitation is that the improvement of our approach on prediction models is dependent on the task and the environment. If the configurability of the environment is limited, the effect size of our algorithm will likely be small. Future work should explore beyond objective measures of human predictability, investigating the subjective implications of our approach on the human's experience (i.e. do more complex configurations increase cognitive load for the human?). Lastly it would be valuable to evaluate our approach over extended interactions to determine whether virtual obstacles remain necessary after multiple interactions or if humans naturally adapt to more legible trajectories after "training".

ACKNOWLEDGMENTS

This work was supported by the Office of Naval Research under Grant N00014-22-1-2482 and the Army Research Laboratory under Grant W911NF-21-2-0126.

REFERENCES

[1] Shray Bansal, Rhys Newbury, Wesley Chan, Akansel Cosgun, Aimee Allen, Dana Kulić, Tom Drummond, and Charles Isbell. 2020. Supportive actions for manipulation in human-robot coworker teams. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11261–11267.

[2] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. 2018. Anticipating Many Futures: Online Human Motion Prediction and Generation for Human-Robot Interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 4563–4570. https://doi.org/10.1109/ICRA.2018.8460651

[3] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems* 32 (2019).

[4] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

[5] Anca D. Dragan, Kenton C.T. Lee, and Siddhartha S. Srinivasa. 2013. Legibility and predictability of robot motion. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 301–308. https://doi.org/10.1109/HRI.2013.6483603

[6] Muhammad Fahad, Zhuo Chen, and Yi Guo. 2018. Learning How Pedestrians Navigate: A Deep Inverse Reinforcement Learning Approach. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 819–826. https://doi.org/10.1109/IROS.2018.8593438

[7] Matthew Fontaine and Stefanos Nikolaidis. 2021. Differentiable quality diversity. *Advances in Neural Information Processing Systems* 34 (2021), 10040–10052.

[8] Matthew C Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. 2021. On the importance of environments in human-robot coordination. *arXiv preprint arXiv:2106.10853* (2021).

[9] Matthew C Fontaine and Stefanos Nikolaidis. 2022. Evaluating Human–Robot Interaction Algorithms in Shared Autonomy via Quality Diversity Scenario Generation. *ACM Transactions on Human-Robot Interaction (THRI)* 11, 3 (2022), 1–30.

[10] David Fridovich-Keil, Andrea Bajcsy, Jaime F Fisac, Sylvia L Herbert, Steven Wang, Anca D Dragan, and Claire J Tomlin. 2020. Confidence-aware motion prediction for real-time collision avoidance. *The International Journal of Robotics Research* 39, 2-3 (2020), 250–265. https://doi.org/10.1177/0278364919859436 arXiv:https://doi.org/10.1177/0278364919859436

[11] Himanshu Gupta, Bradley Hayes, and Zachary Sunberg. 2022. Intention-Aware Navigation in Crowds with Extended-Space POMDP Planning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 562–570.

[12] Sarah Keren, Luis Pineda, Avigdor Gal, Erez Karpas, and Shlomo Zilberstein. 2017. Equi-reward utility maximizing design in stochastic environments. *HSDIP* 2017 (2017), 19.

[13] Anagha Kulkarni, Sarath Sreedharan, Sarah Keren, Tathagata Chakraborti, David E Smith, and Subbarao Kambhampati. 2020. Designing environments conducive to interpretable robot behavior. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10982–10989.

[14] Przemyslaw A. Lasota and Julie A. Shah. 2015. Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human–Robot Collaboration. *Human Factors* 57, 1 (2015), 21–33. https://doi.org/10.1177/0018720814565188 arXiv:https://doi.org/10.1177/0018720814565188 PMID: 25790568.

[15] Przemyslaw A. Lasota and Julie A. Shah. 2017. A multiple-predictor approach to human motion prediction. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2300–2307. https://doi.org/10.1109/ICRA.2017.7989265

[16] Joel Lehman and Kenneth O. Stanley. 2011. Evolving a Diversity of Virtual Creatures through Novelty Search and Local Competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (Dublin, Ireland) *(GECCO '11)*. Association for Computing Machinery, New York, NY, USA, 211–218. https://doi.org/10.1145/2001576.2001606

[17] Qinghua Li, Zhao Zhang, Yue You, Yaqi Mu, and Chao Feng. 2020. Data Driven Models for Human Motion Prediction in Human-Robot Collaboration. *IEEE Access* 8 (2020), 227690–227702. https://doi.org/10.1109/ACCESS.2020.3045994

[18] Ren.C Luo and Licong Mai. 2019. Human Intention Inference and On-Line Human Hand Motion Prediction for Human-Robot Collaboration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5958–5964. https://doi.org/10.1109/IROS40897.2019.8968192

[19] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. 2016. Goal Set Inverse Optimal Control and Iterative Replanning for Predicting Human Reaching Motions in Shared Workspaces. *IEEE Transactions on Robotics* 32, 4 (2016), 897–908. https://doi.org/10.1109/TRO.2016.2581216

[20] Gustav Markkula and Mehmet Dogar. 2022. How accurate models of human behavior are needed for human-robot interaction? For automated driving? *arXiv preprint arXiv:2202.06123* (2022).

[21] Julieta Martinez, Michael J. Black, and Javier Romero. 2017. On Human Motion Prediction Using Recurrent Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[22] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015). https://arxiv.org/pdf/1504.04909.pdf

[23] Meinard Müller. 2007. Dynamic time warping. *Information retrieval for music and motion* (2007), 69–84.

[24] Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 866–875.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[26] Stefania Pellegrinelli, Henny Admoni, Shervin Javdani, and Siddhartha Srinivasa. 2016. Human-robot shared workspace collaboration via hindsight optimization. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 831–838. https://doi.org/10.1109/IROS.2016.7759147

[27] Mark Pfeiffer, Ulrich Schwesinger, Hannes Sommer, Enric Galceran, and Roland Siegwart. 2016. Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2096–2101. https://doi.org/10.1109/IROS.2016.7759329

[28] Claudia Pérez-D'Arpino and Julie A. Shah. 2015. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 6175–6182. https://doi.org/10.1109/ICRA.2015.7140066

[29] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. 2020. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research* 39, 8 (2020), 895–935.

[30] Thomas B Sheridan. 2016. Human–robot interaction: status and challenges. *Human factors* 58, 4 (2016), 525–532.

[31] Rainer Storn and Kenneth Price. 1997. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* 11, 4 (01 Dec 1997), 341–359. https://doi.org/10.1023/A:1008202821328

[32] Aaquib Tabrez, Matthew B Luebbers, and Bradley Hayes. 2020. A survey of mental modeling techniques in human–robot teaming. *Current Robotics Reports* 1 (2020), 259–267.

[33] Yi-Shiuan Tung, Kayleigh Bishop, Bradley Hayes, and Alessandro Roncone. 2022. "Bilevel Optimization for Just-in-Time Robotic Kitting and Delivery via Adaptive Task Segmentation and Scheduling". *IEEE International Conference on Robot & Human Interactive Communication* (2022).

[34] Yi-Shiuan Tung, Matthew B. Luebbers, Alessandro Roncone, and Bradley Hayes. 2023. Improving Human Legibility in Collaborative Robot Tasks through Augmented Reality and Workspace Preparation. In *6th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions (VAM-HRI)*. ACM.

[35] Andrey Vakunov, Chuo-Ling Chang, Fan Zhang, George Sung, Matthias Grundmann, and Valentin Bazarevsky. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. https://mixedreality.cs.cornell.edu/workshop.

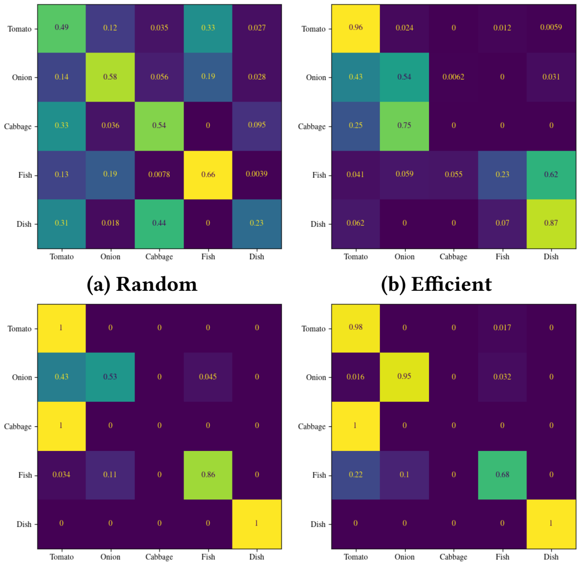
[36] Milos Vasic and Aude Billard. 2013. Safety issues in human-robot interactions. In *2013 IEEE International Conference on Robotics and Automation*. 197–204. https://doi.org/10.1109/ICRA.2013.6630576

[37] Dominik Widmann and Yiannis Karayiannidis. 2018. Human Motion Prediction in Human-Robot Handovers based on Dynamic Movement Primitives. In *2018 European Control Conference (ECC)*. 2781–2787. https://doi.org/10.23919/ECC.2018.8550170

[38] Haoqi Zhang, Yiling Chen, and David C Parkes. 2009. A general approach to environment design with one agent. In *Twenty-First International Joint Conference on Artificial Intelligence*. Citeseer.

[39] Yulun Zhang, Matthew C. Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. 2023. Multi-Robot Coordination and Layout Design for Automated Warehousing. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Edith Elkind (Ed.). International Joint Conferences on Artificial Intelligence Organization, 5503–5511. https://doi.org/10.24963/ijcai.2023/611 Main Track.

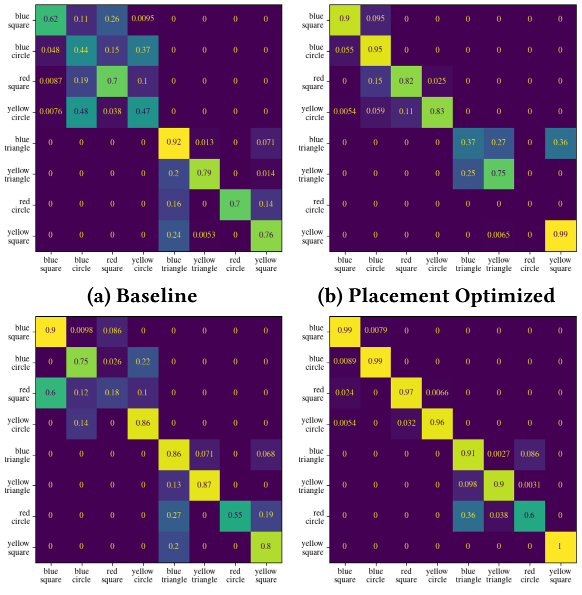
[40] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *AAAI*, Vol. 8. Chicago, IL, USA, 1433–1438.



(a) Random (b) Efficient

(c) Legible + Efficient (d) Legible

Figure 10: Confusion matrix for the Bayesian predictor in the different generated Overcooked layouts.



(a) Baseline (b) Placement Optimized

(c) Virtual Obstacle Optimized (d) Both Optimized

Figure 11: Confusion matrix for the time series multivariate Gaussian in the different generated workspace configurations. The x-axis shows the true target, and the y-axis is the predicted target.

A TASK GRAPH

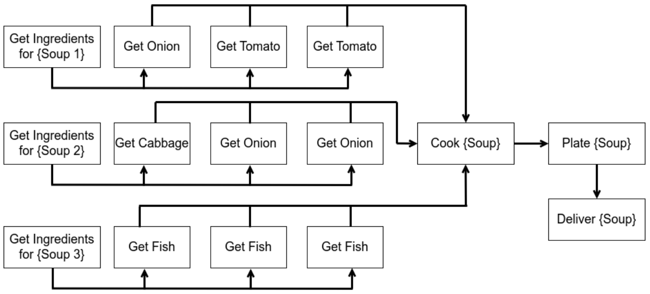
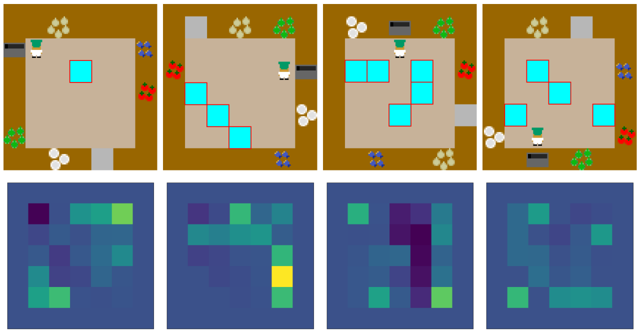


Figure 8: Task graph for the Overcooked game environment. Arrows indicate precedence constraints such that if $t_i \rightarrow t_j$ then subtask t_i has to be completed before subtask t_j can begin. A total of 3 soups have to be delivered, and each soup consists of 3 ingredients. The soups and ingredients can be delivered or placed in any order.

B LEARNED REWARD FUNCTIONS



(a) Random (b) Efficient (c) Legible + Efficient (d) Legible

Figure 9: Learned reward functions from MaxEnt IRL.

We use MaxEnt IRL to estimate the reward functions for each layout. We use the negative of the reward function as the cost function in Equation 1 and predict the target goal of trajectories in the test data set. Figure 9 shows the learned reward function normalized to values between 0 (dark color) and 1 (bright color) for one of the cross validation splits. As expected, the squares near workstations have higher rewards. We can also observe some preferred paths such as taking the right most lane to go from the pot to onion station in the *Legible-Efficient* condition.

C GOAL PREDICTION RESULTS

Figure 10 shows the confusion matrix for the goal prediction across the different conditions in Overcooked. The x-axis shows the true target, and the y-axis is the predicted target. The layout generated by our approach, *Legible*, has 100% correct predictions for the target goals tomato and dish. Due to the environment and task complexity, the optimization trades off the ability to predict the goal cabbage for accurate predictions of the other target goals. Cabbage is only used once in the recipes and thus incurs less cost for incorrect predictions. Note that we only include the results when a partial trajectory is observed since the goal prediction is always correct given the entire trajectory.

Table 2 and 3 show the F1 score, precision, and recall of the goal predictions of the Bayesian predictor in Overcooked and the time series multivariate Gaussian in the tabletop experiments. We

Table 2: Overcooked Goal Prediction

	Random	Efficient	Legible + Efficient	Legible
F1	0.477	0.512	0.744	**0.799**
Precision	0.564	0.648	0.792	**0.824**
Recall	0.482	0.554	0.751	**0.812**

Table 3: Tabletop Goal Prediction

	Baseline	Placement Optimized	Virtual Obstacle Optimized	Both Optimized
F1	0.704	0.753	0.796	**0.936**
Precision	0.751	0.770	0.803	**0.941**
Recall	0.689	0.761	0.800	**0.937**

account for label imbalance by computing the metrics for each label and computing their average weighted by support. We use the implementation from scikit-learn [25].

D COMPARISONS WITH OTHER OPTIMIZATION METHODS

We compared MAP-Elites to Novelty Search with Local Competition (NSLC) [16] and Differential Evolution (DE) [31] for the Overcooked game. We report the average legibility scores across 10 runs with different seeds (larger values are better) in Table 4.

This result shows that MAP-Elites outperforms NSLC and DE, but the best choice of optimization method is likely conditioned on the task and the environment. In the implementation, we set the hyperparameters such that they have the same number of iterations performing the update step for an individual from the population. MAP-Elites initializes the solution map with 100 random layouts and performs the improvement phase for 100 iterations. NSLC maintains a population size of 20 that evolves for 10 generations. DE performs a maximum of 200 generations over which the entire population is updated.

Table 4: Comparisons of optimization methods on the best solution found (i.e. legibility scores) for different Overcooked environments.

	Square 5x5	Square 7x7	Square 9x9	Rectangle 9x5
DE	-129.28	-202.52	-229.22	-186.31
	(2.19)	(3.96)	(4.68)	(3.27)
NSLC	-95.44	-175.43	-190.54	-169.63
	(10.73)	(15.92)	(11.89)	(20.39)
MAP-Elites	**-86.37**	**-167.88**	**−173.52**	**-162.97**
	(7.55)	**(16.13)**	**(8.20)**	**(10.60)**