A Initializing a Pool of Trajectories

Our approach suggests beginning with a pool of trajectories, however obtaining this pool can pose a bootstrapping problem. It's unlikely that a diverse corpus of trajectories—one that would span the feasible attribution space and enable a data-driven extraction approach—would already exist. In order to create such a corpus, one would need to be confident that they have included trajectories that adequately cover the range of attributions that are perceivable in a domain, but the lack of such information is why we want the corpus of trajectories to begin with.

We broke this cycle by referencing existing literature on the perceptions of robot vacuum cleaners and selecting six adjectives as candidate attributions: curious, broken, energetic, lazy, lost, scared. We then manually demonstrated six trajectories that we judged to maximally express these candidates. We collected responses to these trajectories and conducted an initial analysis of what aspects of the motion users said contributed to their ratings. This informed the creation of an initial subset of the features we would use. Additional trajectories were then generated using a hill-descending search in the space of trajectories optimizing for incrementally altering individual features ($\pm$0.1-0.3) selected at random while holding other features constant. We found that alternative optimization criteria, like diversity, would lead to degenerate or trivial trajectories which happened to have extreme feature values. These new trajectories were posed to users for their ratings. Simultaneously, we asked them to demonstrate how they would "clean the bedroom in a way that makes the robot look _____" for random items from our questionnaire. Responses fed back into analysis and the process was cycled through two more times with a subset of the generated and demonstrated trajectories, resulting in the final exploratory dataset and trajectory featurization.

B Loadings

The factor loadings of our final model, derived via exploratory factor analysis of the exploratory dataset, are shown in Table 4.

Variable	Factor 1	Factor 2	Factor 3	Communality
Responsible	**1.00**	.06	−.07	1.01
Competent	**.94**	.03	.00	.89
Efficient	**.93**	.05	−.03	.87
Reliable	**.85**	−.05	.07	.73
Intelligent	**.85**	−.05	.06	.73
Focused	**.84**	−.07	.02	.71
Lost	−.03	**.90**	.01	.80
Clumsy	.13	**.76**	.06	.59
Confused	−.21	**.76**	.03	.63
Broken	−.17	**.62**	−.11	.43
Curious	−.04	.06	**.91**	.83
Investigative	.12	−.02	**.78**	.62

Table 4: Factor loading matrix

C Data Sensitivity

We investigated the sensitivity of our candidate models' performance on unseen trajectories to increased amounts of training data. We divided our final dataset, holding out 20% of the trajectories, then trained and tested models using varying percentages of the training data, repeating the evaluation with 8 randomized folds of the training data. The results, shown in Fig. 6, indicate diminishing returns beginning around the use of 50% of the training data.

D Trajectory Optimization

We would like to generate trajectories that elicit a particular attribution according to the optimization (1). We have the forward model $f_{\boldsymbol{B}|\Xi}|\phi(\xi)$ which predicts a distribution over attributions given

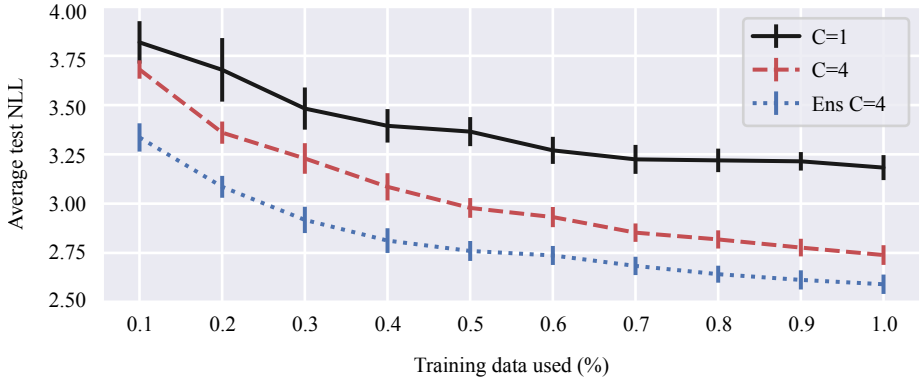


Figure 6: Average test negative log likelihood as a function of the amount of labeled human data used in training. Each datapoint represents the mean of the average test NLL over 10 random folds. Error bars denote standard deviation.

the featurization of a particular trajectory. This model affords two routes towards realizing the optimization. In the first, we directly optimize the features to maximize the objective. However, in that case, a further optimization would be needed to find a trajectory that has the desired feature vector, and it is possible (and common in practice) that there does not exist a trajectory that produces the target features. The second route, which we adopt, is to search in the space of trajectories. This space is large, and in general there is no clear best way to structure the search. Fortunately the features we use have a clear relationship with patterns of motion, so we can easily sample trajectories that increase the activation of individual features. Given a trajectory, we sample a large set of neighboring trajectories using the following modifications:

Action modification : The trajectory is scanned and new trajectories are initialized by individually witholding each state s_i. The single removed state is replaced by selecting every valid alternative action from s_{i-1} and reconnecting the trajectory with a shortest path to s_{i+1}.

Shortcutting : Sections of the trajectory are deleted uniformly at random and the trajectory is reconnected with a shortest path. We sample twice as many cuts as there are states in the trajectory. Shortcutting is a well established post-processing step in motion planning, helping to uncover shorter or cheaper trajectories that still satisfy the objective.

Template Insertion : A sequence of states is patched into the trajectory beginning at each s_i. We used a "straight" template, formed by taking the action used in s_{i-1} and repeating it twice, and a "U" template, formed by taking three actions in proceeding clockwise or counterclockwise directions. The trajectory is reconnected with a shortest path to s_{i+1}. Both templates were directly motivated by participant feedback highlighting these patterns as contributing to the appearance of an organized principle to the robot's motion.

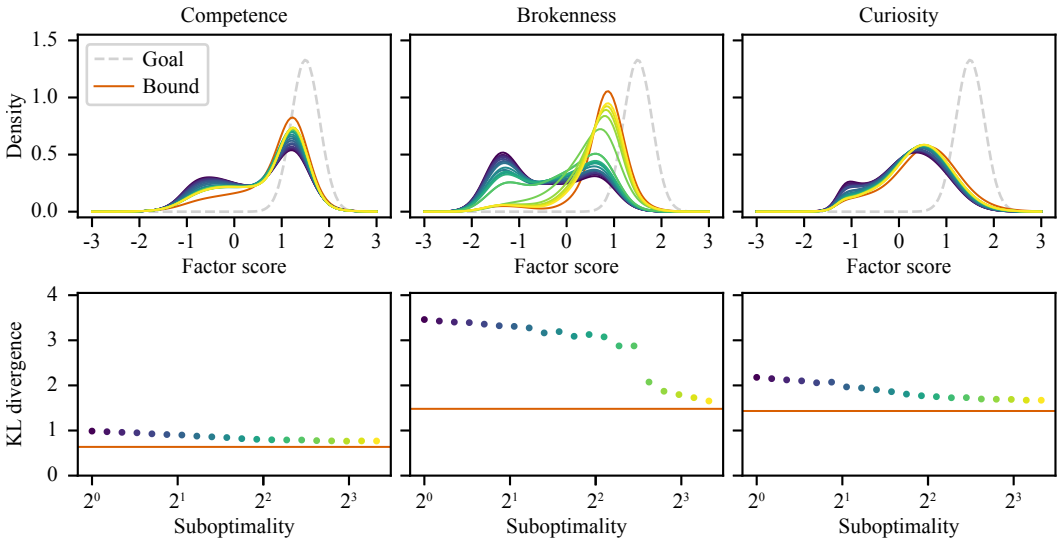
Collision Seeking : The trajectory is scanned and collision-avoiding actions are individually replaced with collision-causing actions if they are available from a state s_i. The trajectory is then reconnected with a shortest path to s_{i+1}. This sampler was motivated by participant responses highlighting collisions as contributing to their ratings of components of the brokenness score. These same modifications are generated by the "Action modification" sampler. We double-sample these to reduce the chance that they are dropped in a subsequent subsampling step.

Overcoverage : A random section of the trajectory $s_i..s_j$ between 3 and 6 states long is selected uniformly at random and the trajectory is modified to backtrack this portion by inserting $reverse(s_i..s_j)$ followed by $s_i..s_j$ in place of s_j. This modification is motivated by participant feedback that highlighted redundant coverage as evidence of attentiveness, exploration or attention to detail.

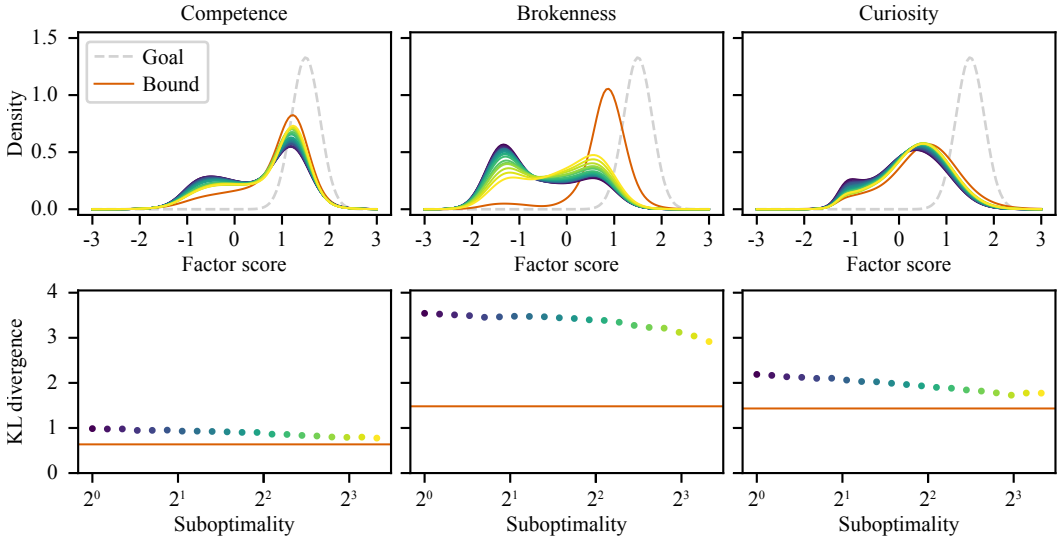
These samplers, and a good initialization produced by solving the original task using a planner, make sampling-based greedy hill-descending search in the space of trajectories effective. As practical considerations, we discard sampled trajectories longer than 250 states to prevent the search from

exploding, then we randomly subsample 250 of the modified trajectories, and we terminate the search once 750 steps have been taken.

As illustrated in Fig. 7, this search process is generally effective at making use of additional allowable suboptimality to produce trajectories that the model predicts to better elicit a desired attribution. Some non-monotonicity can be observed in one of the brokenness and one of the curiosity plots. As trajectories get longer, the pool of neighbors sampled is diluted with many more small changes and our subsampling step can lead the search to randomly discard more fruitful paths.



(a) Training environment (as used in Experiment I)



(b) Modified environment (as used in Experiment II)

Figure 7: The predicted distribution of factor scores for trajectories optimized under varying settings of w, represented as a ratio of the task-optimal cost. The optimization goals are configured as they are in our experiments, a Gaussian centered at 1.5 with standard deviation 0.3, and the plots show the marginal density for the factor under consideration. The companion plots show the KL divergence for each of the plotted distributions, providing a measure of closeness to the goal distribution for each generated trajectory. The bound line represents the distribution and KL divergence that are predicted for a feature vector obtained by optimizing the features directly, as opposed to optimizing in the space of trajectories. In general, these features may not be realizable with a trajectory that obeys the domain constraints, so the predicted distribution represents an upper bound.

E Statistical Results

We conducted one-sample Kolomogorov-Smirnov tests comparing the empirical distribution of factor scores elicited from participants against the distribution predicted by our model. To account for the increased likelihood of Type-I errors due to multiple testing, the Holm-Sidak adjustment was applied to the resulting p values. The results of the tests are provided in Table 5.

	Experiment I						Experiment II					
	Competence		Brokenness		Curiosity		Competence		Brokenness		Curiosity	
	$D(24)$	p	$D(24)$	p	$D(24)$	p	$D(24)$	p	$D(24)$	p	$D(24)$	p
1x	.381	**.014**	.292	.255	.233	.659	.343	.059	.247	.464	.319	.143
2x	.190	.789	.185	.789	.120	.879	.309	.145	.151	.836	.299	.206
4x	.275	.334	.232	.659	.144	.879	.182	.738	.256	.446	.254	.464
12x	.277	.334	.217	.695	.198	.789	.246	.464	.211	.601	.143	.836

Table 5: Results of Kolmogorov-Smirnov tests comparing predicted and observed distributions for each condition

	Experiment I						Experiment II					
Competence		Brokenness		Curiosity		Competence		Brokenness		Curiosity	
τ_b	p	τ_b	p	τ_b	p	τ_b	p	τ_b	p	τ_b	p
-.261	**.001**	.402	**<.001**	-.069	.370	-.175	**.045**	.351	**<.001**	-.033	.675

Table 6: Results of Kendall's tau-b correlation between optimization parameter w and factor score

For Experiment I, the tests failed to indicate a significant difference between the distributions for all but the Competence-1x condition, suggesting that there is insufficient evidence to support inequivalence between predicted and observed distributions. For Experiment II, the tests failed to indicate a significant difference between the distributions for all conditions, suggesting that there is insufficient evidence to support inequivalence between predicted and observed distributions.

We conducted Kendall's tau-b correlations to determine the relationship between the allowable suboptimality of a generated trajectory (1, 2, 4 or 12) and the observed factor scores for the attribution under study. The correlation coefficients and p values are reported in Table 6. The results indicate strong positive correlations between the allowable suboptimality and brokenness factor scores, suggesting that the trajectory generation method was effective at eliciting progressively higher factor scores. However, tests for the competence conditions indicated a moderate negative correlation, and tests for curiosity conditions were not significant.