

Learning the Right Abstraction for Norm Discovery

Daniel Little and Matthias Scheutz

Abstract—Norm learning from demonstrations typically assumes that the representation used to express norms is fixed in advance, even though the relevant representation may itself be uncertain. We formulate norm discovery as joint inference over a normative abstraction and a norm, using an LLM to propose executable abstraction–norm hypotheses and Bayesian maximum-entropy inference to evaluate how well they explain demonstrated behavior. When the current hypothesis set cannot adequately account for new observations, the learner can refine the representation by introducing higher-level state properties, actions, or events. We evaluate the approach in a deterministic shopping gridworld with existing social norms. GPT-5.5 achieves 0.93 ± 0.14 accuracy on classifying held-out behavior that requires generalization beyond the initial low-level representation, compared with 0.333 for a fixed-representation MLCI baseline. These results support treating representation selection as an explicit part of norm discovery rather than assuming the appropriate normative representation is known in advance.

I. INTRODUCTION

Robots operating in human environments are often required to comply with social norms in addition to completing their task objectives. For example, a robot operating in a supermarket could be expected to avoid spilled liquids or respect the personal space of other shoppers, all while navigating the store and searching for particular items on the store shelves. However, several issues make it difficult to determine what social norms are in fact being followed in an environment. The environment could contain a multitude of norms, in which case it is particularly helpful to observe the behavior of other agents in the same domain, potentially humans or other artificial agents, and infer the norms being followed from such examples of normative behavior, called *learning from observation*. However, learning from observation is complicated by the potentially large number of objects, properties, relations, and predicates available to describe each observed state. Therefore, agents that learn norms from observations (henceforth *norm-learning agents*) require a method for identifying the representation relevant to the normative behavior.

A norm-learning agent may initially perceive a rich description of its environment. In a supermarket domain, for example, an observed state might specify whether an agent possesses an item, whether the item has been paid for, whether the agent is at the checkout or exit, what clothing the agent is wearing, what objects are nearby, and many other properties. However, the information relevant to a norm such as “an agent carrying unpaid goods must visit the checkout before exiting” may be more naturally represented

using higher-level concepts that combine multiple lower-level states, predicates, or actions. Learning over the entire state-space can therefore introduce unnecessary complexity, while an under-specified representation may make the true norms impossible to express.

Norm learning therefore also requires choosing an appropriate representation. Existing approaches to symbolic norm or constraint induction typically assume that the features over which constraints are learned have already been specified. Yet the relevant representation may itself be uncertain. Moreover, limited demonstrations can support multiple plausible representations. If every observed shopper wearing a hat also pays before leaving, for example, both payment status and hat-wearing may initially appear to be normative. Only subsequent observations that break this correlation reveal which is normatively relevant.

We therefore consider learning the norm and learning its relevant representation as a joint inference problem. We assume that trajectories are initially available in a symbolic representation containing potentially relevant and irrelevant predicates. A large language model (LLM) uses the semantics of these predicates and the observed behavior to propose candidate *normative abstractions*: higher-level concepts defined over lower-level states, predicates, relations, and actions, over which candidate obligations and prohibitions can be expressed. Rather than accepting the LLM’s judgment directly, we treat each abstraction and corresponding norm as a normative hypothesis whose ability to explain observed behavior can be evaluated probabilistically as additional demonstrations are received.

We perform evaluation with Bayesian inference using a maximum-entropy model of demonstrator behavior. The learner maintains uncertainty over different normative hypotheses and updates their posterior probabilities as new demonstrations arrive. When a new trajectory is poorly predicted by the current hypothesis set, this provides evidence that the current representation or norm is inadequate and triggers LLM-guided refinement of the normative hypotheses.

II. RELATED WORK

Our work is most closely related to inverse reinforcement learning and inverse constraint learning, which infer hidden objectives or behavioral constraints from demonstrations. Classical inverse reinforcement learning seeks a reward function under which demonstrated behavior is likely or near-optimal [1], [2], while Bayesian IRL maintains a distribution over possible reward functions and updates it from observed behavior [3]. Inverse constraint learning instead asks which

constraints could explain why a demonstrator avoided otherwise preferable behavior. Scobee and Sastry [4] compare demonstrated trajectories with cheaper alternatives under the known task objective. When a cheaper alternative is not observed, this suggests that some part of that alternative may violate an unknown constraint. They use a maximum-entropy model to evaluate candidate constraints according to how well they explain the demonstrator’s preference for the observed trajectory over those alternatives. Malik et al. [5] formulate inverse constrained reinforcement learning as the problem of recovering an unknown constraint function from expert demonstrations. They assume the demonstrator is optimizing a known reward while remaining within an unknown feasible region, and infer constraints that make the demonstrated behavior feasible while excluding otherwise high-reward alternatives, while Chou et al. [6] infer constraints shared across demonstrations using grid-based and parametric representations. Later work relaxes some of the assumptions on the demonstrations themselves. CoCoRL [7] considers demonstrations generated under different and potentially unknown reward functions, and confidence-aware ICRL [8] quantifies uncertainty in inferred constraints and whether the available demonstrations provide sufficient evidence.

A second line of work learns explicit symbolic norms or constraints from behavior. Scheutz and Little [9] incrementally infer simple deontic constraints from demonstrations, including low-level obligations and permissions. Baert et al. [10] combine inverse reinforcement learning with inductive logic programming (ILP) to recover interpretable logical constraints. Deane and Ray [11] similarly use ILP to generalize numerically inferred constraints into first-order rules that transfer across environment configurations. Oldenburg and Tan [12] use approximate Bayesian rule induction to infer candidate obligations and prohibitions from observed behavior in multi-agent settings. These approaches are especially relevant because they produce explicit symbolic rules rather than representing the learned structure only through a reward function, however, the representation or language in which candidate norms are expressed is specified in advance.

There is also prior work on learning the representation used for inverse learning. Choi and Kim [13] use Bayesian nonparametric inference to construct composite reward features from predefined atomic features for inverse reinforcement learning. In our setting, the learned representation instead consists of higher-level state properties, actions, and events over which explicit obligations and prohibitions can be expressed.

Related work in planning has also considered how symbolic representations can be learned rather than provided manually. Konidaris et al. [14] construct symbolic representations from low-level state spaces for abstract high-level planning. Silver et al. [15] learn predicates from demonstrations for bilevel planning, and Shah et al. [16] learn symbolic relations between objects from demonstrations for use in long-horizon planning. Foundation models have also been used for this purpose. PDDL-ART [17], for example, uses

a vision-language model to generate PDDL representations from an expert demonstration and a natural-language task description.

III. PROBLEM FORMULATION

A. Normative Planning Domains

We consider a task environment represented by a planning model

$$\mathcal{M} = \langle S, A, T, C \rangle, \quad (1)$$

where S is a set of states, A is a set of actions, $T : S \times A \rightarrow S$ is a deterministic transition function, and $C : S \times A \rightarrow \mathbb{R}$ is a positive task cost. A trajectory is

$$\tau = [\langle s_0, a_0 \rangle, \dots, \langle s_{n-1}, a_{n-1} \rangle], \quad (2)$$

where $s_{k+1} = T(s_k, a_k)$, with task cost

$$C_{\text{task}}(\tau) = \sum_{k=0}^{n-1} C(s_k, a_k). \quad (3)$$

We assume that each state can be described using a symbolic predicate vocabulary

$$\Phi = \{\phi_1, \dots, \phi_m\}. \quad (4)$$

The complete representation may contain information relevant to task execution but irrelevant to the norms being learned. For example, Φ may contain predicates describing whether an agent possesses an item, whether the item has been paid for, whether the agent is at a checkout, and whether the agent is wearing a hat.

B. Normative Abstractions

A *normative abstraction* α consists of higher-level state properties and actions constructed from the low-level representation available to the learner. We write

$$\alpha = (\alpha_S, \alpha_A), \quad (5)$$

where α_S defines abstract state properties and α_A defines abstract actions or events.

A state abstraction constructs higher-level properties from low-level state predicates and relations. For example, an agent’s coordinates together with the locations of surrounding shelves may be used to define the higher-level property `inAisle`. Thus, states with different low-level descriptions may share the same normatively relevant property.

An action abstraction similarly constructs higher-level actions or events from the low-level action representation. An abstract action may correspond directly to a lower-level interaction or may summarize a sequence of lower-level actions. For example, several primitive actions involved in acquiring a shopping cart may be represented by the event `pickUpCart`, while the actions involved in returning it may be represented by `returnCart`.

Applying the abstraction to a low-level trajectory

$$\tau = [\langle s_0, a_0 \rangle, \dots, \langle s_{n-1}, a_{n-1} \rangle] \quad (6)$$

produces an abstract trajectory

$$\bar{\tau} = \alpha(\tau), \quad (7)$$

containing the abstract state properties, actions, and events identified along the trajectory.

A candidate norm N is expressed over the state properties, actions, and events defined by the current abstraction α . Norms may constrain individual state properties or relationships among actions and events. For example, a prohibition may state that an agent may not simultaneously possess a shopping cart and satisfy the predicate `inAisle`. An obligation may instead require that, after `pickUpCart` occurs, `returnCart` must subsequently occur. The abstraction therefore determines the concepts available for expressing a norm, while the norm specifies the normative relationships among those concepts. Given a low-level trajectory τ , the norm is evaluated on the corresponding abstract trajectory $\bar{\tau} = \alpha(\tau)$ to determine whether the demonstrated behavior satisfies or violates the norm.

We define a normative hypothesis as

$$h = (\alpha, N). \quad (8)$$

The objective is therefore not just to infer a norm N , but to identify both a norm and an abstraction sufficient to express it. Given a set of demonstrated trajectories D , we seek to determine which normative hypotheses $h = (\alpha, N)$ best explain the observations.

Rather than assuming that the appropriate abstraction must be proposed from scratch, we assume that the learner begins with an initial abstraction α_0 . This abstraction may reflect the representation naturally available to the learner before normative reasoning begins. For example, the initial state representation may retain raw coordinates rather than a higher-level property such as `inAisle`, while the initial action representation may collapse several different object interactions into a common `INTERACT` action.

The learner first attempts to explain demonstrated behavior using norms defined over this initial abstraction. If the resulting hypothesis set cannot adequately explain the demonstrator's choices, the abstraction itself is refined by an LLM. Refinement may introduce new state properties, or new actions or events.

C. Bayesian Abstraction and Norm Inference

Let $\mathcal{H} = \{h_1, \dots, h_K\}$ denote the current set of candidate normative hypotheses. We maintain a posterior over this set according to

$$P(h \mid D, \mathcal{M}, \mathcal{H}) \propto P(D \mid h, \mathcal{M})P(h), \quad (9)$$

Because $h = (\alpha, N)$, we write

$$P(h) = P(N \mid \alpha)P(\alpha). \quad (10)$$

Here, $P(\alpha)$ captures prior belief about the abstraction used to represent normative behavior, $P(N \mid \alpha)$ represents how plausible we consider a norm before evaluating the demonstrations, and $P(D \mid h, \mathcal{M})$ measures how well the resulting normative model explains observed behavior.

Because different abstractions may vary substantially in how complicated they are to define, we place a prior on α that favors shorter descriptions:

$$P(\alpha) \propto \exp(-\lambda L(\alpha)), \quad (11)$$

where $\lambda \geq 0$ controls the preference for simpler abstractions and $L(\alpha)$ is the description length of the abstraction. Because multiple norms may explain the same observations, we also place a prior on N that favors simpler norms:

$$P(N \mid \alpha) \propto \exp(-\gamma L(N)), \quad (12)$$

where $L(N)$ is the description length of the norm.

For a hypothesis $h = (\alpha, N)$, let

$$V_h(\tau) \geq 0 \quad (13)$$

represent the number or severity of violations of N induced by the abstract trajectory $\alpha(\tau)$. We separate this amount of violation from its cost by introducing $w \geq 0$, which represents the cost the demonstrator assigns to one unit of violation.

We model the demonstrator using a maximum-entropy trajectory distribution [2]:

$$P(\tau \mid h, w, \mathcal{M}) = \frac{\exp[-\beta(C_{\text{task}}(\tau) + wV_h(\tau))]}{Z(h, w, s_0, g)}, \quad (14)$$

where β represents how rational demonstrators are and $Z(h, w, s_0, g)$ is the partition function over feasible trajectories connecting the demonstrated start state s_0 to goal g :

$$Z(h, w, s_0, g) = \sum_{\tau' \in \mathcal{T}(s_0, g)} \exp[-\beta(C_{\text{task}}(\tau') + wV_h(\tau'))]. \quad (15)$$

A hypothesis receives greater support when the observed trajectories are likely under the combination of task cost and norm-violation cost that it defines. Hypotheses that assign high normative cost to behavior inconsistent with the demonstrations can therefore be distinguished from hypotheses that provide a poorer explanation of the observed behavior.

We treat w as unknown and maintain a posterior over both the hypothesis and its violation cost:

$$P(h, w \mid D, \mathcal{M}) \propto P(D \mid h, w, \mathcal{M})P(w \mid h)P(h). \quad (16)$$

In practice, we approximate inference over the continuous violation cost w using a finite grid $\mathcal{W} = \{w_1, \dots, w_J\}$. The posterior is therefore represented as a discrete distribution over pairs (h, w_j) :

$$P(h, w_j \mid D, \mathcal{M}) \propto P(D \mid h, w_j, \mathcal{M})P(w_j \mid h)P(h). \quad (17)$$

Integrals over w in subsequent predictive calculations are correspondingly evaluated as sums over $\mathcal{W}$.

Assuming demonstrations are conditionally independent given (h, w) ,

$$P(D \mid h, w, \mathcal{M}) = \prod_{\tau \in D} P(\tau \mid h, w, \mathcal{M}). \quad (18)$$

Two hypotheses can therefore remain behaviorally consistent while receiving different posterior support when one

explains the demonstrations better. Uncertainty can also persist when the available observations do not distinguish between different normative abstractions.

IV. LLM-GUIDED ABSTRACTION DISCOVERY

A. Hypothesis Generation

We use an LLM to propose possible normative abstractions and corresponding norms. The LLM receives the available predicate vocabulary Φ , natural language descriptions of the predicates and actions, and symbolic representations of observed trajectories. It is asked to identify higher-level concepts that could plausibly explain the behavior in the demonstrations and to propose candidate obligations or prohibitions expressible using those concepts. These concepts may be built hierarchically from lower-level predicates, relations, and actions. This produces a set of normative hypotheses

$$\mathcal{H} = \{(\alpha_1, N_1), \dots, (\alpha_K, N_K)\}. \quad (19)$$

The LLM is used only to propose plausible hypotheses, not to determine which normative hypothesis is accepted. This avoids the need to enumerate every possible abstraction and every possible norm. Each proposal is then evaluated according to how well it explains the observed trajectories using Eq. (16).

The LLM returns each hypothesis in JSON format, specifying both the abstraction and the norm expressed over it. We check that each proposal can be evaluated using the available low-level representation. Applying a hypothesis $h = (\alpha, N)$ to a trajectory produces the violation measure $V_h(\tau)$ used in Eq. (14).

B. Prediction and Updating

As additional trajectories are observed, the posterior is updated sequentially. Given the posterior after observations $D_{1:t}$, a new trajectory τ_{t+1} updates each hypothesis according to

$$P(h, w \mid D_{1:t+1}) \propto P(\tau_{t+1} \mid h, w, \mathcal{M}) P(h, w \mid D_{1:t}). \quad (20)$$

New observations can distinguish between hypotheses that explained earlier demonstrations equally well. In particular, observations may favor one abstraction when the norm defined over it provides a better explanation of the demonstrated behavior.

Let $\mathcal{H}_t$ denote the current hypothesis set after observing $D_{1:t}$. We additionally compute the posterior predictive probability of each new observation:

$$P(\tau_{t+1} \mid D_{1:t}, \mathcal{M}) = \sum_{h \in \mathcal{H}_t} \int P(\tau_{t+1} \mid h, w, \mathcal{M}) P(h, w \mid D_{1:t}, \mathcal{M}) dw. \quad (21)$$

Equation (21) allows us to distinguish between two cases. The learner may be uncertain because several existing hypotheses explain the observations well, in which case no new representation is needed. Alternatively, a new trajectory may be unlikely under all current hypotheses, suggesting that none of the current abstractions and norms adequately explains the behavior.

C. Overall Algorithm

Algorithm 1 summarizes the complete learning procedure. The learner begins with a predefined initial abstraction α_0 and first attempts to explain the demonstrations using norms expressed over that abstraction. The initial LLM call is restricted to the state properties, actions, and events made available by α_0 and proposes candidate norms without modifying the representation. This produces an initial hypothesis set

$$\mathcal{H} = \{(\alpha_0, N_1), \dots, (\alpha_0, N_K)\}. \quad (22)$$

The learner then maintains a posterior over these hypotheses and their associated violation costs as demonstrations are observed.

For each new demonstration, the learner computes its posterior-predictive probability under the current hypothesis set before incorporating it into the posterior. We maintain a running baseline b equal to the smallest predictive probability among previous demonstrations that did not trigger refinement. Abstraction refinement is triggered when

$$p_t < \rho b, \quad (23)$$

where

$$p_t = P(\tau_t \mid D_{1:t-1}, \mathcal{M}) \quad (24)$$

and $\rho = 0.1$ in our experiments. A sufficiently low predictive probability indicates that the current abstractions and norms provide a poor explanation of the newly observed behavior.

When refinement is triggered, the LLM is given the demonstrations in the lower-level representation underlying the current abstraction. It is asked to identify distinctions that were hidden by the current abstraction and use them to construct a refined abstraction. For example, several lower-level interactions initially represented by the single event INTERACT may be separated into distinct events such as pickUpCart and returnCart. The LLM then proposes candidate norms expressed over the refined abstraction. Each proposal is checked against the supported abstraction and the posterior is recomputed using all demonstrations observed so far. If refinement is not triggered, the existing posterior is updated directly using the new demonstration.

V. EXPERIMENTAL EVALUATIONS

A. Experimental Setup

We evaluate the proposed approach in a deterministic shopping gridworld containing shelves, items, a shopping cart, and an exit. We construct two experiments designed to test different forms of normative abstraction. The first requires abstraction over temporally related actions, while the second requires constructing a higher-level state property from lower-level spatial coordinates. In both experiments, demonstrated trajectories satisfy a hidden social norm. We intentionally use a small number of demonstrations to evaluate the framework in a sparse-evidence setting, where the available behavior may not uniquely identify the intended norm.

Algorithm 1 LLM-Guided Normative Abstraction Discovery

Require: Planning model $\mathcal{M}$, demonstrations $D = \{\tau_1, \dots, \tau_T\}$, initial abstraction α_0 , low-level vocabulary Φ

Ensure: Posterior over normative hypotheses and violation costs

```
1:  $\mathcal{N}_0 \leftarrow \text{PROPOSENORMS}(\alpha_0, \tau_1)$ 
2:  $\mathcal{H} \leftarrow \{(\alpha_0, N) : N \in \mathcal{N}_0\}$ 
3: Initialize prior  $P(h, w)$  over  $h \in \mathcal{H}$  and violation costs  $w$ 
4: Compute  $P(h, w \mid \tau_1, \mathcal{M})$  using Eq. (16)
5: Initialize predictive baseline  $b \leftarrow \emptyset$ 
6: for  $t = 2, \dots, T$  do
7:    $p_t \leftarrow P(\tau_t \mid D_{1:t-1}, \mathcal{M})$  using Eq. (21)
8:   if  $b \neq \emptyset$  and  $p_t < \rho b$  then
9:      $\mathcal{H}_{\text{new}} \leftarrow \text{REFINEABSTRACTION}(\Phi, D_{1:t}, \mathcal{H})$ 
10:     $\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{H}_{\text{new}}$ 
11:    Recompute  $P(h, w \mid D_{1:t}, \mathcal{M})$  over  $h \in \mathcal{H}$  using Eq. (16)
12:   else
13:     Update  $P(h, w \mid D_{1:t}, \mathcal{M})$  using Eq. (20)
14:     if  $b = \emptyset$  or  $p_t < b$  then
15:        $b \leftarrow p_t$ 
16:     end if
17:   end if
18: end for
19: return  $P(h, w \mid D, \mathcal{M})$ 
```

We compare five conditions. *Fixed abstraction* uses the initial representation and Bayesian inference procedure but does not permit abstraction refinement. We additionally compare against an implementation of the maximum-likelihood constraint inference (MLCI) method of Scobee and Sasttry [4], which infers constraints over the available low-level state and action representation without constructing new abstractions. Our complete method is evaluated with two LLMs: Qwen2.5-7B-Instruct and GPT-5.5. Finally, the *oracle abstraction* condition provides the correct abstraction at the start while retaining the same Bayesian inference procedure. This provides an upper bound on performance when the appropriate abstraction has already been identified.

For both LLM conditions, all experimental settings other than the choice of LLM are held fixed. We use temperature 0.2, $\beta = 2$, $\lambda = 0.2$, and $\gamma = 0.2$. These values were selected heuristically and held fixed across both experiments rather than tuned separately for each domain. We approximate the posterior over the violation cost using

$$\mathcal{W} = \{0, 0.5, \dots, 30\}. \quad (25)$$

We use a uniform prior over the discrete violation-cost grid, such that

$$P(w_j \mid h) = \frac{1}{|\mathcal{W}|}, \quad w_j \in \mathcal{W}, \quad (26)$$

for every normative hypothesis h . We run each LLM condition for ten independent runs.

A held-out trajectory is classified as violating when the highest-posterior normative hypothesis assigns it at least one norm violation. We report overall held-out classification accuracy and violation F1.

B. Experiment 1: Cart-use Norm

The first experiment evaluates whether LLM-guided refinement can recover an obligation that cannot be expressed adequately using a low-level action representation. The initial abstraction represents all object interactions using the single action `INTERACT`. Consequently, actions such as acquiring a cart, retrieving a food item, and leaving a cart are initially indistinguishable.

The data consists of three demonstrations in which the demonstrator acquires a cart, collects the required items, returns the cart to a designated return area, and then exits. The demonstrations vary in which items are collected, with one demonstration requiring two items.

We evaluate the learned norm on ten held-out trajectories. Four are norm-compliant trajectories containing different items and item orders. The remaining six contain three kinds of norm violations: exiting without returning the cart, leaving the cart at the wrong location, and returning the cart but then picking it up again before exiting.

C. Experiment 2: Aisle-use Norm

The second experiment evaluates whether the learner can construct a state property that generalizes across different low-level spatial coordinates. The hidden social norm is that agents are forbidden from taking their cart into the area between store shelves, where doing so could block other shoppers from accessing the space. However, the initial representation contains only the agent’s coordinates and whether it has a cart. The concept of being in an aisle is therefore not initially available for constructing norms.

The first two demonstrations involve agents traveling to two different aisles in the supermarket. In each case, the demonstrator leaves the cart outside the aisle, enters without it to collect an item, returns to the cart, and then continues to the exit. Because the two aisles occur at different coordinates, a rule tied only to the coordinates from the first demonstration does not provide a general explanation of both demonstrations.

The third demonstration shows the agent retrieving an item from a shelf that is not located in an aisle, such as a standalone display. In this case the agent keeps the cart while collecting the item. This provides evidence against a more general rule that having a cart near any shelf is prohibited.

During abstraction refinement, the LLM is given generic geometric relations describing the grid, including whether a location is free, whether it is adjacent to a shelf, whether it is bounded by shelves on opposing sides, and the width of the surrounding free space. It can combine these lower-level spatial relations to define a new state property. The definition of an aisle itself is not provided to the LLM.

Held-out evaluation includes a third aisle that does not appear in any of the demonstrations. The held-out set contains both norm-compliant and norm-violating behavior in

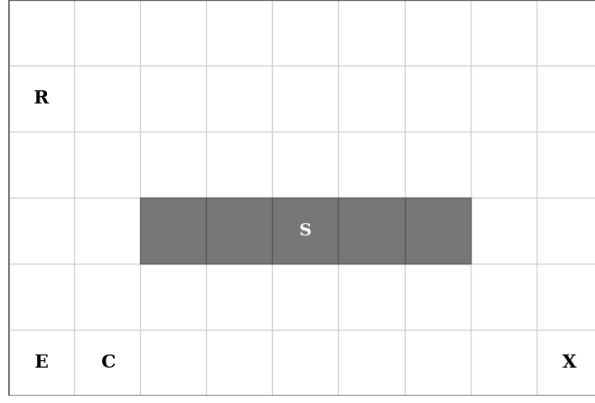


Fig. 1. Shopping domain used in experiment 1. The agent enters the store at location E, picks up a cart at C, retrieves items from the shelves marked S, returns the cart at R, and exits the store at E.

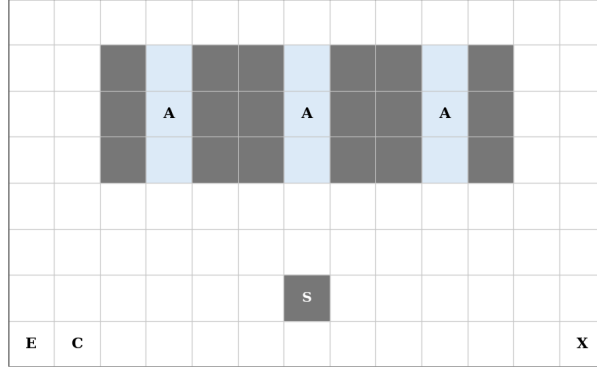


Fig. 2. Shopping domain used in experiment 2. The agent enters the store at location E, picks up a cart at C, retrieves items from the gray shelves where A marks aisles, and exits the store at E.

TABLE I

HELD-OUT NORM CLASSIFICATION RESULTS. LLM RESULTS REPORT THE MEAN $\pm$ STANDARD DEVIATION OVER TEN INDEPENDENT RUNS.

Method	Exp. 1		Exp. 2		
	Acc.	Viol. F1	Acc.	Viol. F1	Unseen Acc.
Fixed abstraction	0.600	0.500	0.200	0.333	0.333
MLCI	0.600	0.500	0.600	0.500	0.333
Ours - Qwen2.5-7B	0.68 ± 0.17	0.60 ± 0.21	0.66 ± 0.28	0.60 ± 0.29	0.57 ± 0.27
Ours - GPT-5.5	0.80 ± 0.00	0.80 ± 0.00	0.96 ± 0.08	1.00 ± 0.00	0.93 ± 0.14
Oracle abstraction	1.000	1.000	1.000	1.000	1.000

TABLE II

ABSTRACTION PROPOSAL AND FINAL MAP SELECTION ACROSS TEN LLM RUNS.

Model	Exp. 1		Exp. 2	
	Proposed	Final MAP	Proposed	Final MAP
Qwen2.5-7B	4/10	2/10	7/10	2/10
GPT-5.5	10/10	0/10	10/10	8/10

this aisle. This tests whether the learned abstraction captures the common spatial structure rather than simply memorizing coordinates that appeared in the demonstrations.

D. Results and Analysis

Table I summarizes held-out behavioral performance. LLM results report the mean $\pm$ standard deviation over ten independent runs. Providing the oracle abstraction yields perfect performance in both experiments, showing that the Bayesian inference procedure can recover the target norm when the required representation is available.

GPT-5.5 improves the generation of useful abstractions, although the effect differs between the two experiments. Table II reports both whether the LLM proposed a correct abstraction and whether that abstraction appeared in the final maximum a posteriori (MAP) hypothesis.

In Experiment 1, the fixed-abstraction and MLCI baselines both achieve an accuracy of 0.600 and violation F1 of 0.500. Qwen2.5-7B improves mean accuracy to 0.68 ± 0.17 and violation F1 to 0.60 ± 0.21 , while GPT-5.5 further improves both metrics to 0.80 ± 0.00 .

Despite this improvement, GPT-5.5 shows that proposing the right abstraction is not enough to guarantee that the right norm is selected. In Experiment 1, GPT-5.5 proposes the intended abstraction in all ten runs, including separate events for picking up and returning the cart. However, the posterior consistently prefers a competing norm in which the obligation to return the cart is triggered by picking up any item rather than by picking up the cart. Both norms fit the three training demonstrations because every demonstration includes both cart pickup and item pickup before the cart

is returned. The difference only becomes visible on held-out trajectories where the cart is returned and then picked up again before exit. The intended norm treats the second cart pickup as creating a new obligation to return it, while the competing norm does not because no additional item is picked up. As a result, the competing norm misclassifies these trajectories, leading to the same 0.80 accuracy across all ten runs. The remaining error therefore comes from the training demonstrations not containing enough information to distinguish the intended norm from this competing explanation, rather than from a failure to propose the appropriate abstraction.

GPT-5.5 performs considerably better in Experiment 2. The fixed-abstraction condition obtains an overall accuracy of 0.200, violation F1 of 0.333, and unseen-structure accuracy of 0.333. MLCI improves overall accuracy to 0.600 and violation F1 to 0.500, but its unseen-structure accuracy remains 0.333. This is consistent with MLCI operating over the supplied low-level representation rather than constructing a new structural state property.

With Qwen2.5-7B, our method achieves an overall accuracy of 0.66 ± 0.28 , violation F1 of 0.60 ± 0.29 , and unseen-structure accuracy of 0.57 ± 0.27 in Experiment 2. GPT-5.5 improves these results to 0.96 ± 0.08 , 1.00 ± 0.00 , and 0.93 ± 0.14 , respectively. GPT-5.5 proposes a correct abstraction in all ten runs, compared with seven of ten Qwen runs, and a hypothesis using that abstraction is selected as final MAP in eight of ten runs, compared with two of ten for Qwen.

VI. DISCUSSION AND FUTURE WORK

Our results suggest that treating the representation used for norm learning as uncertain can provide an advantage over methods that operate only over a fixed set of state and action features. This difference is most visible in Experiment 2. MLCI improves overall accuracy over the fixed-abstraction condition, but its accuracy on held-out behavior involving the previously unseen aisle remains low. In contrast, abstraction refinement allows the learner to construct a state property from relations among shelves and coordinates and apply the resulting norm at locations that were not present in the demonstrations.

Experiment 1 highlights a different limitation. GPT-5.5 proposes a correct abstraction in all ten runs, but a hypothesis using that abstraction is never selected as the final MAP hypothesis. The failure therefore occurs after hypothesis generation. With only three training demonstrations, a competing hypothesis can explain the observed trajectories at least as well under the current likelihood and complexity prior while generalizing less successfully to the held-out cases. This illustrates an important distinction in the framework: an LLM determines which abstractions are available for consideration, while probabilistic inference determines which of those abstractions are supported by the behavioral evidence. Improving the language model proposing abstractions does not by itself resolve ambiguity when the demonstrations do

not sufficiently distinguish competing explanations for the observed behavior.

A further limitation is that the framework depends on the LLM proposing an appropriate abstraction–norm hypothesis. The examples considered here involve norms that are broadly consistent with common human expectations, so the LLM’s prior knowledge may aid hypothesis generation. For unfamiliar or counterintuitive norms, the appropriate hypothesis may not be proposed at all; Bayesian evaluation can reject unsupported candidates, but it cannot recover a hypothesis that never enters the candidate set.

There are several additional limitations to the present evaluation. First, we assume that low-level trajectories are already available in a symbolic representation. Applying the approach to physical robots would require grounding predicates and relations from perception and would introduce uncertainty in the low-level observations themselves. Second, the current experiments use small deterministic planning domains. The maximum-entropy likelihood requires reasoning over feasible trajectories, which becomes substantially more difficult in large, stochastic, or continuous environments. Approximate planning and inference methods will therefore be necessary for larger robotic domains. Third, our experiments consider one primary norm at a time. Real environments may contain several simultaneously active norms whose effects overlap, creating additional ambiguity between normative explanations.

Future work should extend the framework in two directions. First, posterior uncertainty could be used to guide the collection of additional demonstrations that are most informative for distinguishing between competing hypotheses. Rather than relying only on passively observed behavior, the learner could identify situations in which high-probability hypotheses make different predictions and seek additional evidence in those cases. Second, future work should investigate priors and likelihood models that better capture the intended generality of an abstraction and reduce the tendency to favor hypotheses that fit a small number of observations but generalize poorly. These extensions would allow the framework to use uncertainty not only to evaluate candidate norms and abstractions, but also to determine what evidence should be collected next.

VII. CONCLUSION

We presented a framework for norm discovery that treats the representation used to express a norm as part of the inference problem. The learner begins with an initial abstraction, proposes norms over that representation, and refines the abstraction when the current hypothesis set is inadequate. LLM-generated abstraction–norm hypotheses are evaluated using Bayesian maximum-entropy inference rather than accepted directly.

The experiments show why this distinction matters. In Experiment 2, GPT-5.5 reaches 0.93 ± 0.14 accuracy on held-out behavior requiring generalization beyond the initial low-level representation, compared with 0.333 for MLCI. Experiment 1 shows a separate limitation: the intended abstraction

is proposed in every GPT-5.5 run, but a competing norm is selected by the posterior. Norm discovery therefore depends both on making the relevant representation available and on having enough behavioral evidence to distinguish the norms that can be expressed over it.

REFERENCES

- [1] A. Y. Ng and S. Russell, “Algorithms for inverse reinforcement learning,” in *ICML*, 2000.
- [2] B. D. Ziebart, A. Maas, J. Bagnell, and A. Dey, “Maximum entropy inverse reinforcement learning,” in *AAAI*, 2008.
- [3] D. Ramachandran and E. Amir, “Bayesian inverse reinforcement learning,” in *IJCAI*, pp. 2586–2591, 2007.
- [4] D. R. Scobee and S. S. Sastry, “Maximum likelihood constraint inference for inverse reinforcement learning,” in *ICLR*, 2020.
- [5] S. Malik, U. Anwar, A. Aghasi, and A. Ahmed, “Inverse constrained reinforcement learning,” in *ICML*, 2021.
- [6] G. Chou, D. Berenson, and N. Ozay, “Learning constraints from demonstrations with grid and parametric representations,” *The International Journal of Robotics Research*, vol. 40, no. 10–11, pp. 1255–1283, 2021.
- [7] D. Lindner, X. Chen, S. Tschitschek, K. Hofmann, and A. Krause, “Learning safety constraints from demonstrations with unknown rewards,” in *AISTATS*, PMLR, vol. 238, pp. 2386–2394, 2024.
- [8] S. Ganapathi Subramanian, G. Liu, M. Elmahgiubi, K. Rezaee, and P. Poupart, “Confidence aware inverse constrained reinforcement learning,” in *ICML*, PMLR, vol. 235, pp. 14491–14512, 2024.
- [9] M. Scheutz and D. Little, “Using simple deontic constraints for fast norm-conforming reinforcement learning,” in *Proceedings of the 17th International Conference on Deontic Logic and Normative Systems (DEON)*, pp. 329–347, 2025.
- [10] M. Baert, S. Leroux, and P. Simoons, “Inverse reinforcement learning through logic constraint inference,” *Machine Learning*, vol. 112, pp. 2593–2618, 2023.
- [11] O. Deane and O. Ray, “Neuro-symbolic inverse constrained reinforcement learning,” in *Proceedings of the 19th International Conference on Neurosymbolic Learning and Reasoning (NeSy)*, PMLR, vol. 284, pp. 913–925, 2025.
- [12] N. Oldenburg and Z.-X. Tan, “Learning and sustaining shared normative systems via Bayesian rule induction in Markov games,” in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1510–1520, 2024.
- [13] J. Choi and K.-E. Kim, “Bayesian nonparametric feature construction for inverse reinforcement learning,” in *IJCAI*, pp. 1287–1293, 2013.
- [14] D. G. Konidaris, L. P. Kaelbling, and T. Lozano-Pérez, “From skills to symbols: Learning symbolic representations for abstract high-level planning,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 215–289, 2018.
- [15] T. Silver, R. Chitnis, N. Kumar, W. McClinton, T. Lozano-Pérez, L. P. Kaelbling, and J. B. Tenenbaum, “Predicate invention for bilevel planning,” in *AAAI*, vol. 37, no. 10, pp. 12120–12129, 2023.
- [16] N. Shah, J. Nagpal, and S. Srivastava, “From real world to logic and back: Learning generalizable relational concepts for long horizon robot planning,” in *Proceedings of the 9th Conference on Robot Learning (CoRL)*, PMLR, vol. 305, pp. 5362–5434, 2025.
- [17] D. Kamale and D. Berenson, “PDDL-ART: Autonomous symbolic abstraction from demonstration for long-horizon robotic manipulation using vision-language models,” *arXiv preprint arXiv:2608.17146*, 2026.