# A Computational Framework for Norm-Aware Reasoning for Autonomous Systems

Vigneshram Krishnamoorthy, Huao Li, Stephanie Milani, Wenhao Luo, Yue Guo, Michael Lewis, Katia Sycara

**Abstract**

Autonomous agents are increasingly deployed in complex social environments where they not only have to reason about their domain goals but also about the norms that can impose constraints on task performance. The intelligent trade-offs that these systems must make between domain and normative constraints is the key to their acceptance as socially responsible agents in the community. Integrating task planning with norm aware reasoning is a challenging problem due to the curse of dimensionality associated with product spaces of the domain state variables and norm-related variables. To this end, we propose a Modular Normative Markov Decision Process (MNMDP) framework that is shown to have orders of magnitude increase in performance compared to previous approaches. Because norms are both context-dependent and context-sensitive, context must be modeled effectively in order to find activation and deactivation conditions for norms. To this end, we propose an expressive, scalable and generalizable context modeling approach to understand norm activations in social environments combining the expressivity of propositional logic with the compactness of decision trees. We show how we can combine our context model with our MNMDP framework to support norm understanding as well as norm enforcement for real systems. Human experiments were conducted to collect data relating context and norms to populate our framework. We discuss the results and inferences obtained from these data confirming the complexity of the relationship between contexts and norms and the necessity of empirical data to building a scalable norm-aware reasoning framework for autonomous systems. We demonstrate our approach through scenarios in simulated social environments in which agents using the framework display norm-aware behavior. In order to discover which norms get activated in various contexts and with what priorities, we performed a set of extensive human experiments and show how these results constitute the basis for building context-based

norm aware systems.

*Keywords:*
Normative Reasoning, Knowledge Representation, Context Modeling,
Propositional Logic, Decision Trees, Markov Decision Process, SimRank

## 1. Introduction

As autonomous systems become more involved in our daily life, their responsibilities have expanded from industry and military to more personal scenarios like childcare, personal assistant or housekeeping. Instead of only concentrating on the accomplishment of the assignment, autonomous agents have to consider human's expectation and preference in those tasks for them to be accepted and trusted. However, constraints in social interactions are usually fuzzy and soft; thus cannot be explicitly programmed as hand-crafted functions/rules in a unified fashion. One important factor that informs the appropriateness of actions is the set of norms within a particular community. Norms, such as prohibitions, permissions, obligations, are the socially agreed upon guidelines of behavior which are acknowledged by most of the members of a community [1]. With the competence of reasoning about norms, intelligent agents can align their behaviors with human values to better collaborate and communicate with people in a socially desirable way. Thus it is critical for the robots to incorporate social norms in their decision-making process in performing their various tasks.

We are interested in the reasoning of robots engaged in long term autonomy, where they perform a set of tasks in one or multiple domains and when appropriate, engage in normative reasoning to ascertain the consequences of their actions so as to determine their best course of action. There exist few works in the community which aim to tackle this important problem [2, 3]. However, the existing literature suffers heavily from the curse of dimensionality problem and are hence infeasible in real systems. We also tackle the more challenging problem of computational models for reasoning that can be well suited for long-term autonomy applications, where the set of norms that the agent has to follow and the environments that the agent is deployed in, change over time.

Currently, there is little research about generalizable representations and models of environmental context for modeling social interactions. The social science literature [4] [5] have reported that norm activation is dependent on environmental and social context. Additionally, we claim that the determination of priorities among different norms or even between instances of the same norm depends on

the environment where the agent operates. For example in Figure 6, the robot follows the safety norm of saving the child in the context where the child gets too close to a fire before following another instance of the same safety norm which makes the robot extinguish the fire. On the other hand, in the context where the child is away from the fire, the norm precedence changes and the robot's priority is the instance of the safety norm that entails it to put out the fire before saving the child. However, the specific representation of context-dependent norm activation framework in terms of perceiving context, norm enforcement and policy execution is an open problem which this work addresses in detail. Since it is impossible to elicit all possible situations and their norms that the robot will operate under, these systems must be able to learn the mapping between environmental context and norm activations directly from human interaction and also be able to generalize to unseen contexts. On top of the above factors, our work also focuses on scalability, which is the most crucial factor with the choice of the context model in terms of deployment in autonomous systems.

A robot operating openly in society is likely to encounter an immense number of situations that require adherence to differing human norms. If a representation such as an MDP is used to guide behavior this leads to a computationally infeasible solution in which every norm must be considered at every step. In this paper we propose a novel solution in which smaller MDPs incorporating only those norms active in particular contexts are indexed and accessed through a context tree. Section 2 describes Related Work followed by Contributions in Section 3. Section 4 describes a representation for norms. Section 5 introduces the Modular Normative MDP and illustrates its working using a roadway simulator as well as empirical evaluation against previous approaches. Section 6 describes our representation of the environmental context and its organization within our context tree approach. We also demonstrate the implementation of our context model in custom-built Minecraft environments as well as performance studies which illustrate the scalability of our work. Section 7 describes the design of our human experiments used to identify norm priorities and context hierarchies, as well as detailed analysis and results from our experiments.

## 2. Related Work

BOID [6] and NoA [7, 8] present architectures illustrating the potential for norms to be part of the agent behavior. Some authors have studied norm-aware planning algorithms [9, 10, 11], but not from a utilitarian and probabilistic perspective. Further, recent papers [12, 2] propose a framework similar to another

work [13], where Markov Decision Processes (MDPs) are used to combine the agent's domain planning with normative constraints and sanctions, but fail to address the curse of dimensionality problem associated with the large joint state spaces when combining the domain state variables with the normative variables. [14] provides an MDP formulation for scheduling in randomized traffic patrols in a real domain using a game-theoretic approach by modeling the trade-off as a bi-objective optimization problem, but only consider a specific set of interactions between the agents. [15] provide mechanisms for detection and resolution of normative conflicts. A review of approaches for detection and resolution of normative conflicts is presented in [16]. More recently, a computationally scalable framework, called Modular Normative Markov Decision Processes (MNMDPs), was proposed for integrating domain goal and norm reasoning [17].

In order to integrate normative constraints into the domain task planning, there is a need to model and represent social interactions effectively. However, very few works in the literature shed light on scalable, practical, and generalizable systems for modeling social interactions for norm inference. Recent work [5] explores the relationship between environmental context and norms but their formulation doesn't have any explicit context representation and lacks generalizability, and feasibility for real systems. [18] represent norms using deontic logic and learn them under uncertainty from human data but again don't use any explicit context representation for modeling social interactions. In their work, norms are only differentiated based on location and not other potentially important contextual factors (such as the assigned task of the agent), context-sensitive norm activation is ignored, and the computational complexity of the norm-learning algorithm is not scalable. [19] represent normative constraints using Linear Temporal Logic (LTL) and present an approach to determine norm priorities from behavior, which is limited by restrictive assumptions such as considering that the violation cost for norms is only conditioned on the duration of the violation. Additionally, their algorithm computes a product MDP which runs in time exponential to the number of norms, rendering it impractical for real systems.

## 3. Contributions

In this work, we address the hard problem of scalable and generalizable computational models for integrating norm aware reasoning with domain goal planning. We also address this problem from a learning viewpoint, by designing human experiments to enable norm learning and by developing robust learning algorithms and representations that can model and infer from data.

**Scalable Policy Computation and Execution:** A core problem in norm learning and enforcement is to be able to compute robust policies for executing the underlying normative behavior. Norms can be viewed as soft constraints that the agent can violate at the cost of being sanctioned. For example, the optimal execution of the task of going from point A to point B would be to go as fast as one's system can perform. Speed-limits (norm) impose constraints on this 'optimal' agent task execution. Markov Decision Processes (MDP) are a natural way to model the soft trade-offs to be reasoned about by the agent such as whether to be norm compliant and possibly sub-optimal w.r.t the domain task versus executing domain tasks optimally but ignoring norms. Further, MDPs can be extended to stochastic and partially observable domains as well. However, modeling the decision making process directly using MDPs is challenging since it suffers from the curse of dimensionality problem when dealing with large state spaces that come up when we try to add an increasing number of normative variables into the domain MDP that reasons only about reaching the goal. Our Modular Normative Decision Process framework [**?** ] which will be discussed in detail in Section 5 provides a modular, scalable way to compute and execute general normative policies by using the properties of the norms themselves. Its performance both in terms of memory consumption and runtime is orders of magnitudes better than previous approaches as will be discussed in Section 5.2.

**Context Representation and Modeling:** For the MNMDP framework to be successful, it needs the knowledge of which norms are active at any given time. Norms are often sparse in the state-space, but finding when they get activated is crucial for successful norm enforcement. Norms are not only context dependent but are also very context sensitive, making it a challenging problem to predict norm activations just from the environmental context. For example, given the same task in the same location, different norms can get activated depending on other factors such as the characteristic of the people present there (e.g. guest, stranger, owner). Moreover, in many situations, multiple norms can be active at the same time, with some of them being in conflict with one another. In order to resolve these norm conflicts, norm priority/importance must be determined. This calls for a principled way to handle context based activation of a variable number of norms as well as feasible ways to determine norm priorities. However, modeling environmental context in a scalable, generalizable fashion and finding a model that can map it to a set of active norms is an open research problem in the community, which we address in detail. (1) We employ propositional logic to expressively represent environmental context, including context with temporal components; (2) We model the set of environmental contexts in the knowledge

base using learnable decision trees, which provides significant computational benefits and also helps create a hierarchy of relevant context factors; (3) We propose a principled approach for integrating social norms into task planning by combining our context model with the Modular Normative Markov decision process (MN-MDP) framework [17]; (4) We propose extensions to support generalization to unseen scenarios using SimRank [20].

**Learning Norm priorities from human experiments:** To determine the context-dependent norm activation and the ranking of normative priorities to use in our framework, we collected human data using a survey deployed with Amazon Mechanical Turk. This survey is the first study to collect data on attitudes of people on normative behavior of domestic robots and study relative priorities among norms for this domain. We observed important properties of the human norm network including 1) dynamic norm priorities over context, 2) diverse context sensitivities of norms, 3) weighted contextual features in norm activations, and 4) complicated interactions between context features and norms. The method and result of human experiments will be discussed in detail in Section 7.

## 4. Norm Characterization

Ethical principles embody societal values and form the basis of social norms and laws. Norms are complex entities with many different attributes [21]. Norms have *modalities* [22], namely obligations, $\mathcal{O}$, permissions $\mathcal{F}$ and prohibitions $\mathcal{P}$ and they apply to the normative agent, who is termed the *addressee of the norm*. *Sanctions* are imposed on the addressee by the issuing *authority* (e.g. the government), if the norm is violated. A *beneficiary* of a norm is the set of agents, including the addressee that benefit (or suffer) the consequences of norm compliance or violation. Norms also have a *spatial and temporal extent* that could be enforced to a group of addressee agents. An example of this would be time-constrained special permissions that apply to specific regions given to agents in case of a regional emergency.

Norms have a *life-cycle* associated with them while they belong to the knowledge base. The *status* of a norm can be *activated*, *violated*, *contradicted*, *deactivated/expired*, and *obsolete/revoked*. A norm gets activated if its activation conditions fit the current state. Additionally a norm has a *utility* and a *priority*. *Let $E$ be the set of all possible well-formed formulae* comprising first-order predicates over terms (constants, variables and the operators $\wedge$, $\vee$, and $\neg$). Following conventional notation from the normative and MDP literature, we define a norm as follows:

**Definition 1** (Norm Representation) A norm $N$ is represented by a tuple $\langle \nu, \Sigma, \phi_n, \phi_a, \phi_d, \sigma, \delta, \pi \rangle$ where $\nu \in \{\mathcal{O}, \mathcal{P}, \mathcal{F}\}$ denotes the deontic modality and $\Sigma$ is the set of states where the norm applies. The normative context $\phi_n$ is the set of states in $\Sigma$ where $\nu$ applies, depending on the norm modality. Conditions $\phi_a, \phi_d$ denote the activation and deactivation condition respectively, and the sanction $\sigma$ for violating it, where $\sigma$ is a tuple $\langle \beta, \phi_s \rangle$ with $\beta : \Sigma \times A \times \Sigma \rightarrow \mathbb{R}^-$ as the (monetary) penalties, $A$ is the set of actions, and $\phi_s \in E$ is the constraint on actions imposed as a sanction. For example, a moving violation in the traffic domain may incur a monetary penalty and loss of driver licence that restricts future actions of the agent. Lastly, $\delta$ represents the authority that issued the norm and $\pi \in \mathbb{Z}^+$ represents norm priority describing the relative importance among norms.

**Definition 2** (Action-Norm Conflict) An action-norm conflict occurs if an action $a \in A$ of the addressee $\alpha$ contradicts one or more activated norms.

For example, if an action of the robot $\alpha$ is $go\_to\_bedroom$ and bedroom privacy norm which entails robot entry prohibition is activated, the resulting state satisfies $location(\alpha, bedroom)$ which violates the prohibition.

**Definition 3** (Normative Conflict) A normative conflict occurs if two or more activated norms contradict one another. In other words, a conflict arises when a state is simultaneously prohibited and permitted/obliged, and its variables have overlapping values.

For example a norm conflict arises when an activated norm obliges the robot to be in the bathroom while another activated norm prohibits the robot from being in the bathroom. The most common way of resolving normative conflicts is by defining priorities. We have conducted human experiments [23] discussed in detail in Section 7 to identify relationships between norms and contexts in domestic environments and determine priorities among these norms. To resolve norm conflict, the highest priority norm is considered for compliance whereas conflicting lower priority norms get violated.

## 5. Modular Normative MDP

We are interested in designing agents that are *norm-autonomous*, namely agents that reason as to whether to violate or comply with norms. We believe that such agents are more realistic in the context of long-term autonomy, as opposed to agents whose norms are automatically regimented by hard-wiring them into the agent. In one of the most relevant works using MDPs [2], it is assumed that the norm set is invariant and determined at design time. Since norms are invariant, [2]

can do the normative reasoning for all states before runtime. Once norm reasoning is done, the norms are excluded from the framework by encoding only the sanctions from norm violations in the states. This has two limitations: (a) considering all norms (full normative model) at once is extremely computationally intensive, and (b) considering norms to be invariant is unrealistic. Our Modular Normative MDP (MNMDP) framework proposed in [17] alleviates these core problems by (1) allowing for efficient computation of integrated task and normative reasoning by modularizing the full-normative MDP into much smaller MNMDPs, (2) MNMDP allows for efficient addition/removal of norms as the robot engages in *long term autonomy operation and human interaction*.

### *5.1. Markov Decision Process with Norms*

A Markov Decision Process (MDP) with norms is represented by $\mathcal{M} = \langle S, A, R, T, \gamma, N \rangle$, where $S$ denotes the finite set of states, $A$ denotes the finite set of actions, $R : S \times A \times S \rightarrow \mathbb{R}$ is a reward function, $T : S \times A \times S \rightarrow [0, 1]$ is a state-transition function, and $\gamma \in [0, 1]$ is the discount factor. The normative knowledge base $N$ is a set of norms that apply in $\mathcal{M}$.

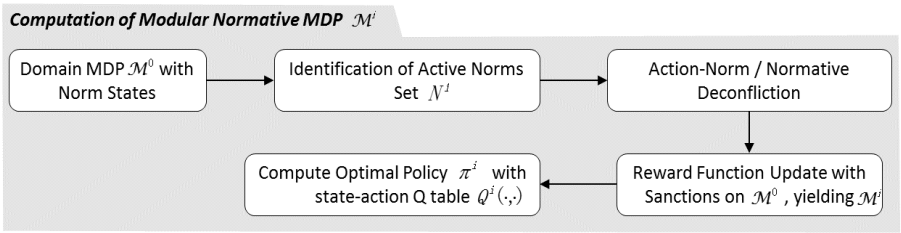



Figure 1: Computation of modular normative MDPs (MNMDPs)

Figure 1 shows the overall process used in computing the policies $\pi^i$ for each MNMDP $\mathcal{M}^i$. We first construct a *domain MDP* $\mathcal{M}^0$, i.e. not including norms in the states. We then use the domain $\Sigma$ of each norm $N$ in the normative knowledge base and compute the MNMDP $\mathcal{M}^{i,j,k}$ when the domain of the norms $N^i$, $N^j$ and $N^k$ overlap and don't conflict with one another. In case of any conflicts between a subset of norms contained in any MNMDP, we resolve it using learned norm priorities derived from our human experiments detailed in Section 7. This modular normative MDP is much smaller than one that would contain the whole norm set. If a norm $N^i$ has no domain overlap with any other norm in the norm set, then this enables us only to compute a MNMDP $\mathcal{M}^i$ which reasons only about the implications of the norm $N^i$ in addition to the task planning. This way, we

pre-compute all the necessary MNMDPs that would be required for the given environment. We then compute the optimal policy using MDP solver algorithms such as value/policy iteration. With $\mathcal{B}^i$ referring to the penalty function for the norm $N^i$, we formulate the reward function $R^i$ for each MNMDP $\mathcal{M}^i$ as

$$R^i(s_t, a, s_{t+1}) = R^0(s_t, a, s_{t+1}) + \mathcal{B}^i(s_t, a, s_{t+1}, N^i) \tag{1}$$

Using this, for each MNMDP $\mathcal{M}^i$ we compute the value function $V$ for each state $s \in S$ as shown in Equation 2.

$$V^i_{k+1}(s_t) = \max_{a \in A^i} \underbrace{\Big[ \sum_{s_{t+1} \in S} P^i(s_{t+1}|s_t, a)(R^i(s_t, a, s_{t+1}) + \gamma V^i_k(s_{t+1})) \Big]}_{Q^i_{k+1}(s_t, a)} \tag{2}$$

until convergence where $k$ refers to the current iteration. Using the converged value function $V$, we compute the optimal MDP policy $\pi^i$ for each MNMDP $\mathcal{M}^i$.

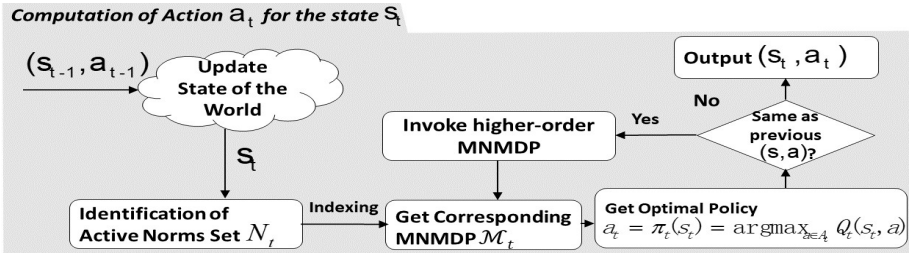
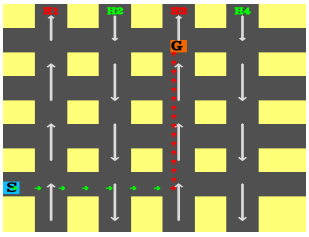
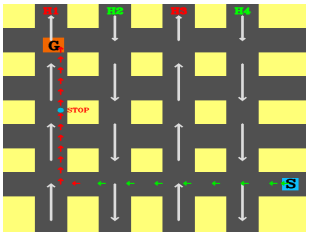



Figure 2: Computation of action $a_t$ for the state $s_t$ during execution

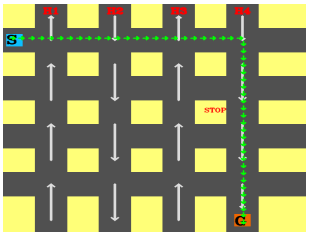
Figure 2 explains the policy execution at runtime, using the pre-computed MNMDP policies as explained above. During policy execution at time $t$ with state $s_t \in S$, let $N_t \subseteq N$ represents the set of activated norms associated with $s_t$. The robot with active norms set $N_t$ will select the best action $a_t$ according to the optimal policy $\pi_t$ for the corresponding MNMDP $\mathcal{M}_t$, so that the robot can achieve domain-specific goals while satisfying the normative constraints. In this way, the robot will only consider the normative constraints in the states where the normative constraints apply, avoiding the unnecessary incorporation of all possible norms in its current MDP $\mathcal{M}_t$. We refer the reader to [17] for further details about the approach.



(a) Direction and Speed-limit norms



(b) Direction, Speed-limit and STOP sign norms



(c) Hospital Emergency, Direction, Speed-limit and STOP sign norms

Figure 3: Trajectories of the agent in different norm constrained environments in a traffic example

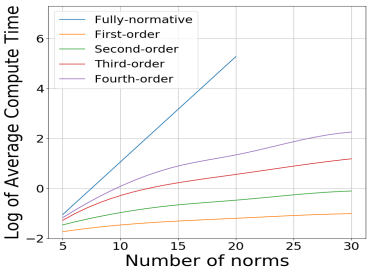
### *5.2. Experimental Results*

#### *5.2.1. Roadway Simulator*

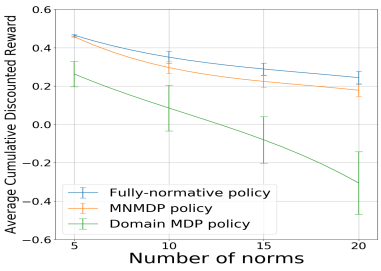
We illustrate the efficacy of our MNMDP framework on a custom-designed roadway simulator encapsulating traffic rules such as speed-limits, STOP signs, and lane directions. The videos for these experiments can be found at this link: http://bit.ly/2GA4HLj. Figure 3 shows the output policies generated by our MNMDP framework for the four different scenarios. Lane directions to be followed are marked using arrows in the four vertical roadways H1-H4 and the speed-limit enforced is shown using red font on the highway signs. In the scenario represented in Figure 7(a) where the agent must follow the speed-limit norm on H1 and H3 in addition to the existing lane norms, the agent follows the speed limits imposed on H3 and slows down. In Figure 7(b), the agent encounters an additional STOP sign norm in addition to the norms discussed in the previous example. It follows the STOP sign placed in H1 as illustrated by the blue circle. In Figure 3(c), the agent encounters a hospital emergency in addition to the lane direction rules, speed-limit on all four vertical highways as well as a STOP sign norm on H4. However, the agent choose to disregard all other norms in favor of reaching the goal as quickly as possible, since the hospital emergency norm (to save a human life) which conflicts with the other norms is given a higher priority.

#### *5.2.2. Empirical Evaluation*

In order to validate the scalability and efficiency of the proposed MNMDP, we compare the performance of our work against a fully normative MDP, which reasons about all the norms together by including all norms in the state space.




(a) Log of computation time against number of norms




(b) Average Cumulative Discounted reward

Figure 4: Comparison of computation time of our MNMDP against the full normative MDP and comparison of cumulative reward of our MNMDP policy, full normative MDP policy and domain MDP policy.

For these experiments, we construct a domain MDP $\mathcal{M}^0$ with a random transition function $T$ and a sparse, random reward function $R$. The number of norms $|N|$ is a parameter that can be varied during evaluation. Each of the norms can be characterized by a fixed number of norm state and action variables, which are both chosen randomly. These settings are used so that we can mitigate any structural bias introduced by analyzing a set of hand-crafted examples, as well as to test the limits of our system with an increasing number of norms.

From Figure 4(a), we can see that $K^{th}$ order interactions take $x^K$ time. For the fully normative MDP, which considers all the norms in the norm set together, this becomes $x^{|N|}$, making it exponential in the number of norms. Figure 4(b) shows the average cumulative discounted rewards for the fully-normative MDP policy, our MNMDP policy, and the domain MDP policy. Note that the domain MDP policy disregards all normative constraints imposed on the system. We observe the general behavior that the cumulative reward decreases as the number of norms increases. We also observe that our MNMDP policy is only slightly sub-optimal to the fully-normative policy, which reasons in a very high-dimensional space, making it intractable for real systems.

## 6. Modeling Environmental Context to Determine Norm Activations

To be socially compliant, agents must determine which norms are relevant, or active, in a given scenario and leverage this information to robustly adapt their behavior. A critical challenge for creating such socially compliant agents is enabling

these agents to use their low-level sensory observations of the environmental context to identify the appropriate normative behavior. This environmental context representation and the mapping from context to norms needs to be generalizable and learnable, especially for long-term autonomy applications where the agent could encounter novel context factors or changing user preferences. In such cases, the framework must scale, be robust enough to generalize to novel environmental contexts, and have the capability to continuously learn and adapt in response to changing user preferences and shifting socio-cultural norms. In this section, we detail our expressive and generalizable computational framework for modeling social context by combining logic-based approaches from artificial intelligence with data-driven, machine learning techniques.

### *6.1. Representing Environmental Context with Propositional Logic*

To represent environmental context, we use propositional logic. With this approach, we model entities in the state, such as the various household objects and attributes, with atomic propositions and the interactions between the entities with logical connectives (e.g. AND, OR). Chaining connectives together enables us to build and express complex, intricate context conditions.

#### *6.1.1. Temporal Logic for Time-Based Context*

Because some of the interactions have a temporal component, we must model the flow of time in our logic. One formal logic that enables this capability is Linear Temporal Logic (LTL), which is a modal temporal logic in which one can define and evaluate propositions over a sequence of states, or paths [24]. A recent extension to LTL, called time window temporal logic (TWTL) [25], was proposed for reasoning about temporal events with more strict time constraints. The TWTL framework enables formalization of tasks with various time constraints, such as performing actions within deadlines, within time windows, in sequence or in strict sequence, and that are enabling conditions for future actions. To handle time-based normative constraints, we utilize the expressivity of both LTL and TWTL for describing various time-bounded contextual variables. We incorporate some of the operators from LTL and TWTL, as well as introduce additional operators - including action-specific, possessive, and spatial operators - for greater expressivity in our context representation.

#### *6.1.2. Propositional Logic for Determining Normative Constraints*

We define our formula $\varphi$ for modeling environmental context over a set of atomic propositions *AP*. Our formula has the following syntax:

$$\varphi ::= H^d s | H^d \neg s | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \neg \varphi_1 | \varphi_1 \cdot \varphi_2 | [\varphi_1]^{a,b}$$
$$| X\varphi_1 | F\varphi_1 | \Box \varphi_1 | \varphi_1 U \varphi_2 | \varphi_1 \triangledown \varphi_2 | \varphi_1 \vdash \varphi_2 | \varphi_1 \dagger \varphi_2$$

where $s$ is either the "true constant" T or an atomic proposition in *AP*; and $\wedge$, $\vee$, and $\neg$ respectively represent the conjunction, disjunction, and negation Boolean operators. The concatenation operator $\cdot$ specifies that first $\varphi_1$ must be satisfied and then immediately $\varphi_2$ must be satisfied. The *hold* operator $H^d$ with $d \in Z_{\geq 0}$ specifies that $s \in AP$ should be repeated for $d$ time units and the *within operator* $[\ ]^{[a,b]}$ with $0 \leq a \leq b$ bounds the satisfaction of $\phi$ to the time window $[a, b]$. The successor (next) operator $X\varphi_1$ specifies that $\varphi_1$ must hold at the next state, the sometimes operator $F\varphi_1$ is the eventually operator modeling the case where the condition $\varphi_1$ is eventually true sometime during the execution, the always operator $\Box\varphi_1$ specifies that $\varphi_1$ must be true in all states, and the until operator $\varphi_1 U \varphi_2$ means $\varphi_1$ until $\varphi_2$. The action operator $\triangledown$ represents an action $\varphi_2$ that is being taken by $\varphi_1$, the descriptive operator $\vdash$ represents "has", and the spatial operator $\dagger$ represents "in" with $\varphi_1$ as the entity and $\varphi_2$ as a location attribute.

We use this logic to express more complex environmental contextual factors for activating or deactivating norms, such as:

- *Context in sequence*, which refers to when one contextual variable must occur after another one. For example, a conversation between a robot's owner and their guest can be modeled as follows:

  ((isOwner() $\triangledown$ isTalking()) $\cdot$ (isGuest() $\triangledown$ isTalking())) $\vee$ ((isGuest() $\triangledown$ isTalking()) $\cdot$ (isOwner() $\triangledown$ isTalking()))

  In other words, to be in a conversation, the owner must talk and then the guest must talk, or vice versa.

- *Context within a time window*, which refers to when a contextual variable must occur within a particular window of time according to the time counter for all agents. For example, person speaking twice within the time window of [0, 5] can be modeled as follows:

  isPerson() $\triangledown$ [H$^2$isSpeaking()]$^{[0,5]}$

  In other words, the person must speak twice within the first 6 time units according to the global time counter.

*6.2. Constructing the Context Tree*

In this section, we describe the structure of our knowledge base, which contains mappings between environmental context and norm activations. A realistic knowledge base for a domestic service robot will contain a wide range of scenarios, where each scenario is represented by a long chain of propositional logic to determine the relevant norms for that scenario. Independently searching through each of these scenarios to determine the relevant norms is highly inefficient. Therefore, we elect to structure our knowledge base as a tree, where nodes represent some context variable(s) that evaluate to either true or false. Like a decision tree, the most differentiating factors are placed near the root of the tree.

We depart from the traditional binary decision tree construction by permitting our context nodes to contain an additional attribute. This attribute corresponds to the set of activated intermediate norms given the evaluation of the evaluation of the parent nodes leading to the current node. This tree structure helps with representation and addresses two major problems that may arise in real robotics systems: (1) partially observable environments and (2) time-critical applications. In partially observable environments, some of the context cannot be evaluated, so by using intermediate norm outputs, the robot can still determine norm activations given the context that it can observe. In time-critical applications, the agent must make a decision about which norms are activated without reaching the leaves of the tree, which is enabled by using intermediate norm outputs. Importantly, note that it is possible to learn these intermediate norm activation outputs using standard tree-learning algorithms.

Figure 5 illustrates a simple example of modeling a scenario into a compact decision tree. The root of this tree evaluates whether the hallway is on fire and if this is True, then the robot checks if the owner is in the hallway. If the owner is present, then the robot checks if the owner is approaching the fire. If so, then the norm activation of preventing the owner from incurring harm enables the robot to save the owner from the fire. On the other hand, if the owner is not approaching the fire or is not present, then the robot focuses on extinguishing the fire. Note that the intermediate norm output at the owner-in-hallway node is the output of the tree based on all the previously traversed nodes - in this case, only the root of the tree. In the other branch of the tree, where the presence of fire evaluates to False, the robot checks for the co-existence of itself and the owner in the hallway. If this is true, the robot determines that the accommodation norm is activated; if it is False, then no norm is activated. In this case, the intermediate output does not enforce any norm since no fire was present and the location of the owner is unknown.

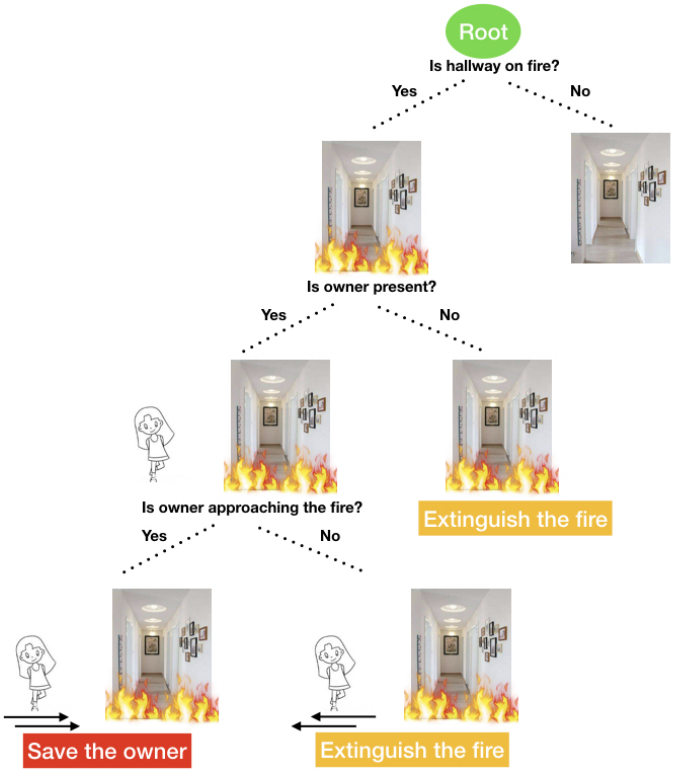



Figure 5: A simple example of modeling a scenario into a compact decision tree. In this example, the yellow boxes are associated with the danger norm; the red box is associated with the protect owner from harm norm.

### *6.2.1. Constructing the Nodes of the Tree*

As previously mentioned, the nodes of our tree represent some contextual variable(s) that evaluate to either true or false. We use the scenarios from recent work [23] as the input for creating our tree-structured knowledge base by, first, mapping each of these scenarios to our chained propositional logic formulae. We split each chained propositional formula based on the occurrence frequency of each sub-part in the chain throughout the set of contexts in our knowledge base. We use this co-occurrence as a measure to derive the features of our context tree so that we can reduce redundancies and hence speed up construction and traversal times. It is intuitive to think of these re-used features as a systematic way to narrow down the search by differentiating between the scenarios where they are present/absent. We set a threshold of interaction frequency $f_t$, which we use to

determine whether an interaction should be separated out into its own node. If an interaction frequency exceeds $f_t$, then it is separated out into its own node. If the current interaction does not occur frequently in the knowledge base, then it remains chained together, as its information gain is low. The threshold $f_t$ is a hyper-parameter that can be empirically tuned for a given application. For example, if the formula "((isOwner() $\triangledown$ isTalking()) $\cdot$ (isGuest() $\triangledown$ isTalking()))" (let's call it owner-guest talking) occurs in a scenario made up of other propositions chained to the above formula, but co-occurence frequency of the above formula in the knowledge base is high, then we can split the chained propositional formula at owner-guest talking not only to promote reuse, but also to help differentiating between all the scenarios with the same owner-guest talking interaction to the ones that don't.

### *6.3. Learning the Context Tree*

Given a set of scenarios as chained propositional formula, we can split them into a set of interaction nodes as discussed in the previous section. Some of these nodes can be much more differentiating than the others. Although the norm output depends on each specific context that the scenario has, there exists an efficient order in which we can check the interaction nodes. For example, the robot should first check the location to see if it is in the bedroom, kitchen, and hallway etc, and then decide which context needs to be examined according to the location. It would not check the food allergy in bathroom, but would only do that in the living room or the kitchen. With a increasing number of scenarios in the knowledge base, a hierarchical structure of context nodes is an intuitive way to model the problem. For a wide variety of general scenarios, we need to be able to learn the most efficient hierarchical representation of the context nodes, such that we can search and retrieve the norms associated with our scenario as fast as possible. Hence, we draw tree learning methods from the machine learning literature that enables us to learn our context trees directly from user data collected in our human experiments [23], as will be explained in detail in Section 7.

We base our tree construction algorithm on the CART [26] framework. To help us place the most differentiating nodes closer to the root node of the tree, we use a multi-variate variant of the Gini impurity measure [27] as our metric. Hence, as we traverse down the tree, we iteratively refine our output norm activation by conditioning it on the current context node that is being evaluated. Each of the leaves of this tree contains the final norm activations for the corresponding paths traversed to reach them.

**Algorithm 1** Traversing a Context Tree

**function** SEARCH-TREE(Observation $O$, Context Tree $C$, Knowledge Graph $K$)
  Node $P \leftarrow$ GET-ROOT(Context Tree $C$)
  Time $T \leftarrow$ INITIALIZE-COUNTER( )
  **while** VALID($P$) **do**
    EVALUATE-NODE($P$, $O$)
    **if** NO-IMPLICATION($P$, $O$) **then**
      $P \leftarrow$ SIMILAR-NODE($P$,$K$)
    **if** IS-NORM($P$) or not OBSERVABLE-NODE(P)
    or $T \geq \tau$ **then return** NORM($P$)
    **else**
      $P \leftarrow$ GET-NEXT-NODE($P$, $C$)
      $T \leftarrow$ INCREMENT-COUNTER( )

We use data from human experiments [23], as will be explained in Section 7, which contain independent priority values for each norm class for each scenario. Thus, we pose our tree learning as a multi-variate regression problem similar to [28]. We create a dataset $D$ containing $Z$ scenarios with each scenario characterized by an input vector of $m$ different propositional functions $X_1, ..., X_m$. We have $x^{(i)} = (x_1^{(i)}, ..., x_m^{(i)})$ as our features for the $i^{th}$ training instance. Our output vectors for the $d$ different norm classes are $Y_1, ..., Y_d$. The output vector for the $i^{th}$ training instance is $y^{(i)} = (y_1^{(i)}, ..., y_d^{(i)})$ with each $y_l^{(i)} \in [1, 7], l \in \{1, ..., Z\}$. We redefine the impurity measure for our multi-variate learning problem by making it the sum of squared error over the multi-variate response for each node,

$$L = \sum_{i=1}^{Z} \sum_{j=1}^{d} (y_j^{(i)} - \bar{y}_j) \tag{3}$$

where $y_j^{(i)}$ denotes the value of the output variable $Y_j$ for the $i^{th}$ instance and $\bar{y}_j$ is the mean of $Y_j$ in the node, with each split selected to minimize this sum of squared error. The computational complexity for constructing this regression tree with $m$ different context features encapsulating $Z$ different scenarios is $O(m\ Z log\ Z)$. We construct this tree offline before policy execution.

*6.4. Efficiently Traversing the Context Tree*

During policy execution, the robot uses its current observation as input to our context tree to efficiently find the norms that are to be enforced, so that the norm-sanctioned policies could be retrieved and executed. Algorithm 1 shows the flow of the tree traversal for any scenario that is part of the knowledge base. Starting from the root, we update the graph edge activation function $\alpha$ using the current observation of the agent. After that, we evaluate the current context node's logic function using $\alpha$. If the current node $P$ has no implication for the given observation $O$, then we use our approach for generalization using the knowledge graph $K$ as explained in Section 6.5 to find the most similar substitute with similar semantics and then continue our traversal from there. Given the structure of our decision tree model, which also stores the intermediate norm outputs, we incorporate the presence of possible time-constraints and/or partial observability to output the norm $N$ without reaching the leaf nodes. Once the branch reaches a leaf node, the search time $T$ is more than a set threshold $\tau$, or the current node $P$ is not observable, we return the predicted norm priorities given the observed context. These norm priorities $\rho$ are then used to resolve any conflicts between norms in our norm set. Note that a set of norms conflict when a state becomes simultaneously prohibited by some norm and permitted/obliged by others. After resolving norm conflicts, we use a set of priority thresholds $\rho_t$ to estimate the value of our norm activation function $\phi_a$ and the deactivation function $\phi_d$ which is modeled as the former's complement. Hence, we find the set of activated norms from the predicted priorities. The search procedure for our context tree is executed at each time step and is $O(log\ n)$, where $n$ is the total number of nodes in the context tree.

Algorithm 2 shows how our context model couples with the MNMDP framework, where the agent's observation is used to obtain a norm-sanctioned policy $\pi$ at each time-step. As explained in Section 6.2, we build our context tree $C$ from the given scenarios $Z$. Then, at each time-step, the agent's observation is used to search through the context tree to obtain the active norms $N$, as explained in Section 6.4. Then, we use the MNMDP framework to find the pre-computed MNMDP policy $\pi$, which reasons about the set of active norms, and execute actions from $\pi$. We repeat this process until we reach a terminal state in the MDP. We show the significant computational advantages of using this approach in Section 6.6.2 compared to a tabular baseline. Note that the knowledge graph $K$ is used to generalize our approach to scenarios outside our knowledge base, as will be explained further in Section 6.5.

**Algorithm 2** Full Pipeline for Determining Agent's Policy

**function** START(Scenarios $S$)
  Graph $K \leftarrow$ CREATE-KNOWLEDGE-GRAPH($S$)
  Context Tree $C \leftarrow$ BUILD-TREE($S$)
  **while** Not-In-Terminal-State **do**
    Observation $O \leftarrow$ GET-OBSERVATION
    Policy $\pi \leftarrow$ GET-POLICY($O$, $C$, $K$)
    EXECUTE-ACTION($\pi$)
**function** BUILD-TREE(Scenarios $S$)
  Context Tree $C \leftarrow$ CREATE-CONTEXT-TREE($S$)
  **return** $C$
**function** GET-POLICY($O$, $C$, $K$)
  Active Norms $N \leftarrow$ SEARCH-TREE($O, C$, $K$)
  Policy $\pi \leftarrow$ GET-MNMDP-POLICY($S$, $N$)
  **return** $\pi$

### *6.5. Generalizing to Unseen Context*

Although our approach of combining propositional logic with decision trees makes for an expressive and efficient framework, it cannot directly generalize to unseen scenarios. One way to achieve generalization is to traverse down the tree to the node where no further implications are present and then use its partial norm output. However, this approach can lead to incorrect inferences of norm activations or deactivations - especially in novel situations.

A better approach is to use similarity metrics to approximate the unknown interaction to some known interaction in a larger knowledge base. To that end, we construct a k-partite interaction graph using the same nodes in the context tree and making dense connections between the nodes based on their co-existence in the scenarios present in our knowledge base. The $k$ different partitions are determined using a heuristic on the categories present in the node (e.g. agent, action, location). By using the aforementioned high-level semantic types to partition the data into a k-partite graph, we restrict the similarity space of a given node to only its neighbours in the same category. The intuition is that nodes containing highly similar semantic information will have common connections in the interaction graph. With this approach, when the traversal procedure encounters a context variable that was previously unseen in a scenario, the knowledge graph is used to retrieve the most similar context variable. To determine the most similar context variable, we use the SimRank similarity measure.

*6.5.1. Using the SimRank Similarity Measure for Generalization*

SimRank is a well-studied, graph-theoretic structural similarity measure applicable in domains with object-to-object relationships [20]. Objects and relationships are modeled as a directed graph $G = (V, E)$, where nodes in $V$ represent objects of the domain, and edges in $E$ represent relationships between objects. For a node $v$ in a graph, the set of in-neighbors and out-neighbors of $v$ are denoted by $I(v)$ and $O(v)$, respectively. Individual in-neighbors are represented as $I_i(v)$, for $1 \leq i \leq |I(v)|$; individual out-neighbors are represented as $O_i(v)$, for $1 \leq i \leq |O(v)|$. SimRank computes similarity scores between nodes using the structural context in which they appear. The similarity $s(a, b) \in [0, 1]$ between objects $a$ and $b$ is 1 if $a = b$, otherwise:

$$s(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) \tag{4}$$

where $C$ is a constant between 0 and 1.

We decompose the scenarios into the key features, and each scenario contains exactly one categorical value for each feature. We construct the k-partite graph by assuming the possible values (aka entities) to be nodes and put a bi-directional edge linking between two nodes if they both appear in a single scenario. Thus for each scenario, there are k edges constructed, where k is the number of features. After processing all the scenarios, some pair of nodes might be linked with more edges than other pairs, which means the entities represented by the two nodes appear more frequently together than others.

As we know for each scenario, there is only one value that can be taken for each feature. It means the edges cannot be formed between two nodes from the same feature, and the nodes from the same feature form groups. As there are k features, we have a k-partite graph. Applying the SimRank similarity metric in Equation 4, we have the similarity metric for each feature accordingly. The similarity metric explains for each feature, how similar each pair of entities are. This similarity is based on the intuition that similar nodes are referenced by the nodes that are also similar. Here we assume if the two entities are linked frequently by similar entities, they have a high score in similarity. Therefore, when given a new scenario unseen before, it is firstly checked through the context tree to the point that it cannot find a reference. Then for this feature that it cannot find a reference, it picks the branch that is the most similar to its entity in this feature. Finally when it is led to the most similar available scenario, our model outputs the same norm activations that are associated with the similar scenario obtained in the

process.

We illustrate our approach for generalization using the following example. Suppose the robot has never observed the guest cooking in the kitchen; however, it has observed its owner cooking in the kitchen. When traversing the context tree, after encountering the location node (kitchen) and the task node (cooking), it encounters its first previously unseen context variable: the guest. Using the SimRank similarity metric in Equation 4 and the knowledge graph, the most similar context variable to the guest is determined - in this case, the robot's owner. The robot's owner then replaces the guest in the context tree, evaluates to true, and the search proceeds to the next context variable or corresponding active norm(s). The active norms for the owner cooking in the kitchen are then used for the scenario of the guest cooking in the kitchen.

### *6.6. Experimental Evaluation*

In this section, we present our results of the implementation of our model for representing environmental context and determining norm activations. First, we deploy our context model coupled with our MNMDP framework as described in Algorithm 2 in custom-built simulation environments emulating complex social norms built on top of Minecraft. This shows the feasibility and effectiveness of our approach for usage in autonomous agents. Then, we proceed to empirical performance studies aimed at determining the computational advantages of using our framework for scalability.

#### *6.6.1. Minecraft*

We use the Project Malmo platform [29] to construct our scenarios and run our experiments. Project Malmo is a platform built on top of the open-world game Minecraft, where researchers can define many diverse, complex problems for intelligent agents to solve. Despite the large number of environments created in Malmo, to our knowledge, no domestic environments currently exist for the platform. To address this, we created four domestic scenarios for testing our context-reasoning model (two of which stem from our survey [23]), which we describe in this section. We use the scenario illustrated in Figure 6, as the primary scenario to illustrate our results.

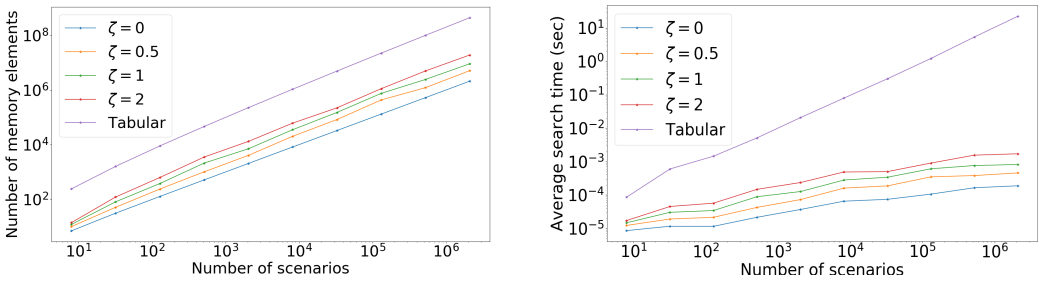
As illustrated in Figure 6, in this scenario the robot is monitoring the kitchen when, suddenly, there is a fire on the stove. At the same time, a child enters the kitchen and either approaches the fire or approaches the window to look outside. As soon as the robot sees the fire, the norm to extinguish the fire is activated. However, when the robot observes the child entering the kitchen, it must decide

between leading the child out of the kitchen and then returning to put out the fire, or ignoring the child and immediately extinguishing the fire.

In our scenario, if the child is far away from the fire, the robot chooses to first extinguish the fire; otherwise, it chooses to first redirect the child away from the fire. After the fire is extinguished, the robot returns to its original location and continues to monitor the kitchen. If there is no fire, the robot stays still and does nothing even if the child comes in and approaches the stove. In this scenario, it seems that the two norms of extinguishing the fire and save the child have conflicts with each other, but indeed the norm of extinguishing the fire has been "cached" when the norm of saving the child has been activated because the latter has the higher priority. However, when the child has been saved and the corresponding norm has been deactivated, the robot resumes the norm of extinguishing the fire based on its observation on the continuing existence of fire. The video of the implementation of this scenario can be found **here** or at https://vimeo.com/319540509.


Figure 6: Example stove scenario in Malmo. The robot (the male figure with black hair) is monitoring the kitchen when, suddenly, a fire starts on the stove in the upper left corner. At the same time, a child (the female figure with blonde hair) enters the kitchen. The robot determines if it should first put out the fire or move the child to a safe area, then put out the fire.

*Implementation details.* In order to obtain information about the objects and agents in the given scene, we tap into Malmo's symbolic observations. This helps us build the observation map containing all relevant interaction details required for our model. We also use the chat box in Malmo, which allows agents to globally communicate messages with each other, to simulate conversations between agents. We add this chat observation and a time counter for our sequential tasks to our observation map. We use the scikit-learn framework [30] for our tree learning. In order to augment the learning process for the scenarios implemented in Malmo, we use the additional heuristic that the two most differentiating context factors are the robot's assigned task and its location. To match the specifications of the scenario, we pre-specify the behavior of other agents (e.g. human agent walking in the hallway) and objects in the environments.




(a) Memory consumption (b) Average search time

Figure 7: Performance comparison between the tabular approach and our proposed context tree approach. Note that both the X and Y axes are in log-scale.

### *6.6.2. Performance Study*

To the best of our knowledge, our work is the first to provide a framework for context modeling for norm aware reasoning, and hence there are no existing benchmarks to evaluate our model against. Hence, we consider a tabular approach with a populated list of scenarios as the baseline against which we can compare our performance against. Our experimental setup for these experiments involves randomizing chains of propositions together to form our interactions. This way we construct our set of scenarios and build our context tree as in Section 6.2.

Using this experiment, we wish to study the scalability of our approach with increasingly unbalanced trees. Consider the mean depth of a given context tree to be $d_\mu$ and the standard deviation of the tree depth to be $d_\sigma$. The coefficient of variation $\zeta = \frac{d_\sigma}{d_\mu}$, which the ratio between the standard-deviation and the mean depth of the tree, capturing the degree of imbalance of the tree for any $d_\mu$ and $d_\sigma$. We try various different $\zeta$ against the number of scenarios to show the scalability trends of our approach. Figure 7 shows the performance comparison between our approach and the tabular approach. Since the context trees become increasingly unbalanced with increasing $\zeta$, the memory and search time increase correspondingly, but still remains close together compared to the baseline. Therefore, the proposed context tree approach is orders of magnitude better than the tabular approach, making it suitable for usage in real systems.

## 7. Human Experiments to learn Normative Preferences

To determine the context-dependent norm activations and the ranking of normative priorities to use in our framework, we collected human data using a survey

deployed with Amazon Mechanical Turk. Because domestic service robots are a vibrant emerging market [31], we focus on scenarios involving domestic robots in a household environment in our survey (note that the MNMDP framework could accommodate a much wider range of contexts and norms). When people interact with domestic robots, their attitudes and acceptance of these robots are influenced by various issues, such as the robot's capability to finish a given task, an invasion of privacy that may arise due to the robot's presence, and numerous potential safety risks [32, 33, 34, 35, 36]. These concerns reveal how people expect domestic robots to behave in typical social interactions. These expectations can, in turn, be aggregated as general values or norms acknowledged by the majority of community members.

### *7.1. Method*

In this section, we detail the methods that we used in our survey. We begin by explaining the design of the scenarios included in our survey, providing a sample scenario for clarity. We then detail our questionnaire design and our data collection approach. We conclude by providing information about the survey participants.

#### *7.1.1. Scenario Design*

As defined in section 4, a norm consists of its *modalities* $\nu$, *activation conditions* $\phi_a$, *penalties* $\sigma$ and *priority* $\pi$ etc. To measure these properties, we tried to design scenarios to survey people's attitude towards the normative behavior of domestic robots. By mapping participants' preference of potential actions and the context in scenarios, we could have a measurement of the norm activation condition as well as the priorities among norms. Inspired by literature on human-robot interaction and ethics, we considered a non-exhaustive list of norms (Table 1) that arise in the interaction between humans and domestic robots. This list provided a guideline for studying people's attitudes and preferences on the appropriate normative behaviors of domestic robots.

We distinguish between the following terms when discussing the survey:

- A *scenario*, or *situation* refers to the description of the setting where the robot must choose how to respond.
- An *option* refers to a possible action that the robot could take in a specific scenario, which indicates the priority of a certain norm.

| Index | Name | Modalities | Description |
|---|---|---|---|
| 1 | Safety | $\mathcal{O}$ | Protect human from danger |
| 2 | Consideration | $\mathcal{O}$ | Consider human's feelings |
| 3 | Privacy | $\mathcal{O}$ | Protect human's privacy |
| 4 | Security | $\mathcal{P}$ | Disclose sensitive information |
| 5 | Efficiency | $\mathcal{O}$ | Finish the given task efficiently |
| 6 | Compliance | $\mathcal{P}$ | Violate social rules |
| 7 | Obedience | $\mathcal{O}$ | Follow owner's command |
| 8 | Command | $\mathcal{P}$ | Act without owner's command |
| 9 | Accommodation | $\mathcal{O}$ | Accommodate human's behavior |
| 10 | Honesty | $\mathcal{O}$ | Tell the truth |
| 11 | Loyalty | $\mathcal{O}$ | Maximize owner's interest |
| 12 | Harm | $\mathcal{P}$ | Harm human |

Table 1: Norm List

- *Context* refers to the physical and social environmental features which are manipulated within a scenario.
- *Norms* are the principal expectations, prohibitions and permissions on the behaviors of domestic robots.

Based on the norm list, we structured *scenarios* where a robot's current action contradicts one or more activated norms (action-norm conflict), or several activated norms conflict with each other (normative conflict) so the robot would need to choose the most appropriate action.

By matching norms from the list, we designed 15 *scenarios*, referred to as *dilemmas*, where two or three norms that the robot should obey conflicted. The robot would have several optional actions to take in the given scenario and each of them corresponds to the addressing of a general norm. For each scenario, participants were required to *independently rate their opinion towards each presented option on a scale of 1 (strongly disagree) to 7 (strongly agree)*. Participants' attitude towards those options reveals the norm priorities and the severity of the conflict between norms in a given context. Both the absolute rating value and relative rank order of options were used to compute the norm priority.

To model the context dependent activation of norms, which refers to the activation and deactivation condition in the definition, we manipulated crucial context variables in the scenario to see if those factors influenced people's attitude. Con-

text was varied in three aspects: 1) location of the interaction, 2) characteristics of those in the scenario, including age, number of people, and their relationship to the domestic service robot owner, and 3) the consequences of robot's behavior which varied in severity. By matching variables with different values, several versions of the scenarios were created, and each participant read a randomly chosen one. By comparing the answers between participants who read different versions of the same scenario, we determined the influence of contexts on the activation and priority of norms.

*7.1.2. Sample Scenario*

To illustrate the notions of scenario, context, and potential normative actions of the robots, we provide a running example below, referenced throughout as the *warning of allergy* scenario:

> *Imagine that you are hosting a dinner at your home.* ***You/your robot*** *finish cooking and are about to serve a dish to your* ***friends/family members****. Your robot detects that one of the guests may be* ***slightly/severely*** *allergic to an ingredient in the dish through the conversation between them. He/she didn't provide allergy information in advance. You and the guests are mid-conversation. Please rate your opinion toward the following statements about robot's behavior.*
>
> - The robot should interrupt the ongoing conversation and tell everyone about the allergy information.
> - The robot should discreetly inform you (the robot owner), but not anyone else.
> - The robot should do nothing.

For example, in the warning of allergy scenario, the dilemma was between three norms *ensure safety*, *consider people's feelings*, and *do not violate social rules*. The robot could interrupt the conversation immediately and inform everyone about the allergy to address the "ensure safety" norm. However, doing so may embarrass the hosts/owners and guests and violate table manners. An alternative action of the robot to better consider everyone's feeling could be discreetly inform the owner of the situation, but not anyone else. As a consequence, it sacrifices the opportunity to immediately warn the guest. The third option, "do nothing due to table manners", emphasizes compliance with social rules like do not interrupt ongoing conversations.

| Index | Scenarios | Norms |
|---|---|---|
| 1 | Reminding events | Security, Obedience |
| 2 | Warning of allergy | Safety, Consideration, Compliance |
| 3 | Entering room | Safety, Consideration, Privacy |
| 4 | Monitoring children | Safety, Privacy, Obedience |
| 5 | Waxing floor | Safety, Efficiency |
| 6 | Interrupting conversation | Consideration, Efficiency, Command |
| 7 | Ordering groceries | Privacy, Efficiency |
| 8 | Encountering human | Safety, Compliance, Accommodation |
| 9 | Organizing room | Privacy, Efficiency, Accommodation |
| 10 | Disposing trash | Privacy, Efficiency, Loyalty |
| 11 | Reporting evidence | Honesty, Loyalty, Command |
| 12 | Judging outfit | Consideration, Honesty |
| 13 | Fire rescuing | Safety, Obedience |
| 14 | Caring elders | Obedience, Consideration |
| 15 | Tackling burglar | Safety, Loyalty, Harm |

Table 2: Scenario List

To measure the activation condition of norms, we manipulated key words in the scenario descriptions to present the scenario with different *context*. In the allergy case, we changed three contextual factors: 1) the agent who is serving the dish is either the owner or robot, indicating the subject of *responsibility*, 2) the *relationship* between the owner and guests is friend or family, and 3) the *danger* of allergy is slight or severe. Variants of scenarios were randomly distributed between subjects. For example, one participant may read the version where the consequence of allergy is mild, but the other may read the one where it is severe.

*7.1.3. Questionnaire Design*

In total 15 scenarios were designed to capture distinct norm conflicts using the method introduced above. Each participant read all of the 15 scenarios; however, the context of each scenario randomly varied between subjects. For example, the allergy case had three contextual variables, each with two different levels that were changed: responsibility, relationship, and the severity of consequence. This gives rise to 8 (2*2*2) different versions of this specific scenario that were evenly distributed to participants. The number of versions for each scenario ranged from 1 to 8 depending on the number and levels of contextual variables. For each scenario, participants were asked to rate their opinion towards every given options

on a 7-point Likert scale (1- strongly disagree and 7- strongly agree). If one option was more favored than other choices, it indicates that the corresponding norm was dominant in the given context. The absolute ratings over all norms are used to map the norm priorities with certain context variables. To help participants better understand the scenario, we provided a detailed description and corresponding picture for each version of scenarios. Pictures were collected from the Internet by us to match the described scenario as well as the context. A sample set of illustrative pictures for the same scenario is shown as Fig. 8, in which the context is different in the location (home or office), number of people present (a group conversation or individual conversation), and relationship between people (friends/colleagues/families). Before the main section, we included demographic information questions to collect participants' gender, age, employment status, educational background, familiarity with robots, and primary source of information for knowledge of robots. After the main section, participants were asked two general questions about their acceptance and willingness to purchase domestic service robots such as those described in the survey.

#### *7.1.4. Data Collection*

To combat the limitations of traditional survey methods, such as less representative sampling [37], we conducted the survey using online platform Qualtrics.com, and published on Amazon Mechanical Turk for workers to access. Each participant was paid $1 to finish the 15-minute survey.

#### *7.1.5. Participants*

To ensure that the participants from Amazon Mechanical Turk were from the American population, we first filtered the submissions by their IP address. Then after removing data of participants that did not finish the survey, or those who blindly clicked and had an extremely short finishing time, and those who answered the attention check question incorrectly, we were left with 301 valid submissions. The sampling population had an average age of 40.6 (SD = 11.6) and the gender ratio was about balanced (49.28% male, 50.71% female).

### *7.2. Results*

A mapping from the contextual variables and norm priorities was constructed for use in the computational norm framework. To give a general description of the collected data, statistical analysis was run to see how contextual features influence the norm priorities. For each norm, we only considered the scenarios in which it was involved and run ANOVA for all the contextual variables. For variables

Figure 8: A sample set of illustrative pictures for the interrupting conversation scenario. Each of them refers to a group conversation among families at home (upper-left), an individual conversation between friends at home (upper-right), a group conversation among colleagues in the office (lower-left), and an individual conversation between colleagues in the office (lower-right). These pictures were presented with corresponding scenario descriptions to help participants understand the scene.

with more than two values, post-hoc tests using Bonferroni method were reported. Since we have in total 11 contextual features and 12 norms, presenting all results from ANOVAs would be less informative for the purpose of model building. Here we only selected the most representative to report, to give a general picture of the human data results.

*7.2.1. Safety norm: Protect human from danger*

The rating of safety norm was significantly influenced by the Domain goals ($F(2, 2085) = 18.90, p < .001, \eta_p^2 = 0.9\%$), Number of people present ($F(2, 2085) = 28.70, p < .001, \eta_p^2 = 2.7\%$), Characteristic of the people ($F(5, 2085) = 124.81, p < .001, \eta_p^2 = 23.0\%$), Importance ($F(1, 2085) = 34.24, p < .001, \eta_p^2 = 1.6\%$), and Emergency ($F(1, 2085) = 22.25, p < .001, \eta_p^2 = 1.1\%$).

Post-hoc comparisons showed that participants thought that the robot should obey the safety norm more when there were people present, $p < .001$. As shown in the left figure of Fig. 9, the ratings of safety norm were higher when the num-

ber of people in context was one or many compared to none. This finding, along with the significant difference found in other variables, indicate the context sensitivity of norm priority. Besides the main effect of each feature, there were also interaction effects between context features on norm priority. The interaction between characteristic and emergency was significant ($F(1, 2098) = 71.39, p < .001, \eta_p^2 = 3.3\%$). For instance in the middle figure of Fig. 9, the level of emergency had little effect on safety norm when there was no specific characteristic for the agent in context, $p > .05$. However, when interacting with children, the robot was expected to consider emergency in the context in order to decide the priority of safety norm, $p < .01$. Similar interaction occurred in the number of people and domain goal factors ($F(1, 2101) = 50.02, p < .001, \eta_p^2 = 2.3\%$), that the effect of domain goals on safety norm was different when the number of people varied (as shown in the right figure of Fig. 9).

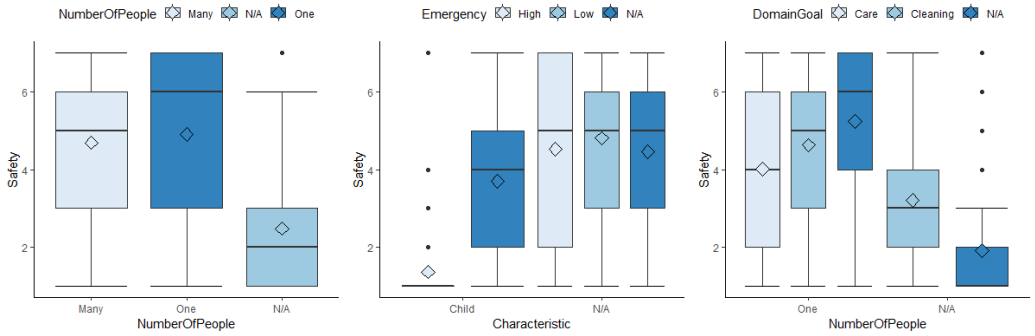



Figure 9: Box plots of the *safety norm* ratings. The upper and lower boundary of box and two error bars represent quartiles of the sample. The line and rhombus within the box indicate the median and mean value of the data respectively. Dots outside the error bars are outliers in the data. The color coding among boxes refers to different variable levels which is shown in the legend. **Left**: Safety norm ratings when *number of people* in the scenario was different. **Middle**: Safety norm ratings when characteristic of agent in the scenario was different. **Right**: Safety norm ratings when *number of people* and robot's *domain goal* were different.

### *7.2.2. Consideration norm: Consider human's feeling*

ANOVA tests showed that context variables Domain goals ($F(2, 1201) = 26.88, p < .001, \eta_p^2 = 4.3\%$), Number of people present ($F(1, 1202) = 44.17, p < .001, \eta_p^2 = 3.5\%$), and Emergency ($F(2, 1201) = 22.00, p < .001, \eta_p^2 = 3.5\%$) significantly influenced the activation of consideration norm.

As shown in Fig. 10, in personal assistant tasks, the robot was not expected to consider people's feeling ($3.95 \pm 2.00$) as in cleaning ($5.00 \pm 1.80$) or non-task situations ($4.80 \pm 1.96$)$ps < .001$. When the number of people present was more than one, the robot was more favored to consider human's feeling ($5.28 \pm 1.79$vs.$4.42 \pm 1.99$) $p < .001$. In high emergency situations, the consideration norm was less valued by the participants ($3.64 \pm 1.94$) than in low ($4.93 \pm 1.86$) or non-emergency situations ($4.75 \pm 1.95$), $ps < .001$. No interaction effect was found in the activation of consideration norm.

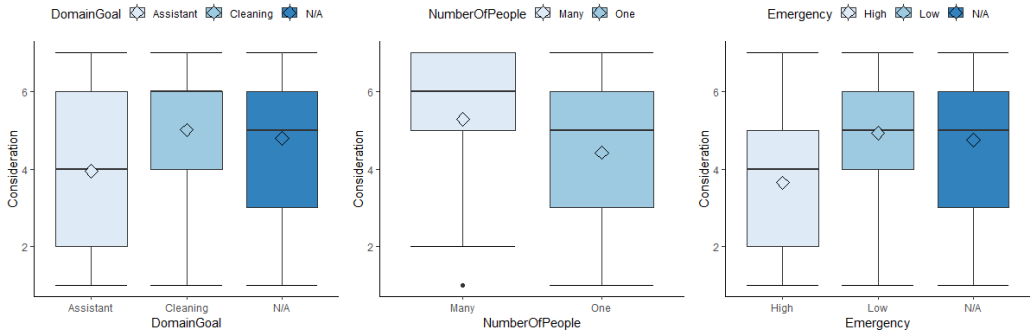



Figure 10: Box plots of the *consideration norm* ratings. **Left**: Consideration norm ratings when robot's *domain goal* was different. **Middle**: Consideration norm ratings when *number of people* in the scenario was different. **Right**: Consideration norm ratings when *emergency* level in the scenario was different.

### *7.2.3. Obedience norm: Follow owner's command*

The influence of characteristics on the obedience norm can be divided into three levels ($F(4, 1199) = 69.14, p < .001, \eta_p^2 = 18.7\%$), the first level is when interacting with children ($6.17 \pm 1.25$), the obedience priority is the highest. Second level is interacting with people in need (elders ($5.10 \pm 1.68$) and disabled people ($5.40 \pm 1.68$)). For the third level of ordinary people without special characteristics the priority of following commands was relatively lower ($4.20 \pm 1.95$). The degree of danger in the environment significantly impact participant's expectation towards the obedience norm. In dangerous situations, people valued the robot to follow owner's command more, compared with normal situations. ($F(3, 1200) = 112.79, p < .001, \eta_p^2 = 22.0\%$, N/A: $3.69 \pm 1.85$, low: $5.56 \pm 1.57$, medium: $5.11 \pm 1.67$, high: $5.92 \pm 1.41$).

When the number of people increased, the importance of following the owner's command decreased ($F(2, 1201) = 89.93, p < .001, \eta_p^2 = 13.0\%$ N/A: $5.66 \pm$

1.50, one: $5.10 \pm 1.86$, many: $3.55 \pm 1.79$). Owner's current task importance also influenced the priority of the obedience norm ($F(2, 1201) = 100.77, p < .001, \eta_p^2 = 14.4\%$, N/A: $5.59 \pm 1.58$, low: $3.68 \pm 1.83$, high: $4.51 \pm 1.95$). Additionally, the interaction between number of people and importance of owner's task was significant ($F(2, 1197) = 26.20, p < .001, \eta_p^2 = 4.2\%$). As shown in the right figure of Fig. 11, the task importance was less influential when there were many people present compared to other situations.

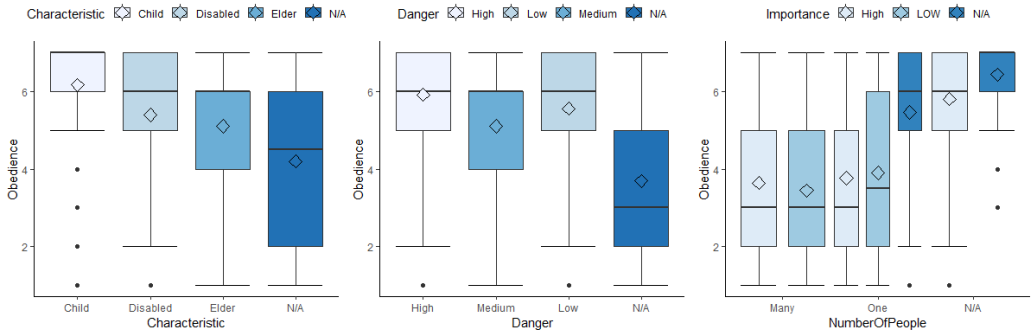



Figure 11: Box plots of the *obedience norm* ratings. **Left**: Obedience norm ratings when characteristic of agent in the scenario was different. **Middle**: Obedience norm ratings when *danger level* in the scenario was different. **Right**: Obedience norm ratings when *number of people* and *task importance* in the scenario were different.

### *7.3. Decision Tree Illustration*

Figure 12 shows the top 3 levels of the constructed decision trees using the categorical context features present in our human experiments data. Danger was found to be the most differentiating feature of our decision tree. The agent can then reason about its domain goal in the case where there is low or no danger. Then if the goal of the agent is to clean, then it takes into account the number of people present in the environment. For tasks such as personal goal and people care, it could start looking at relationship of the people present in the environment. In the presence of danger, the agent then looks at its location where the danger is present. For private locations inside the house such as bedroom, bathroom etc. it investigates the presence of an emergency presumably to reason about privacy related norms. For outdoor locations such as hallway, sidewalk etc., it then looks at the characteristic of the agent such as beggar, pet, elder, children etc. The tree is complex and continues on refining the scenarios further and further until it can differentiate them in terms of the norm activations they entail. The leaf nodes

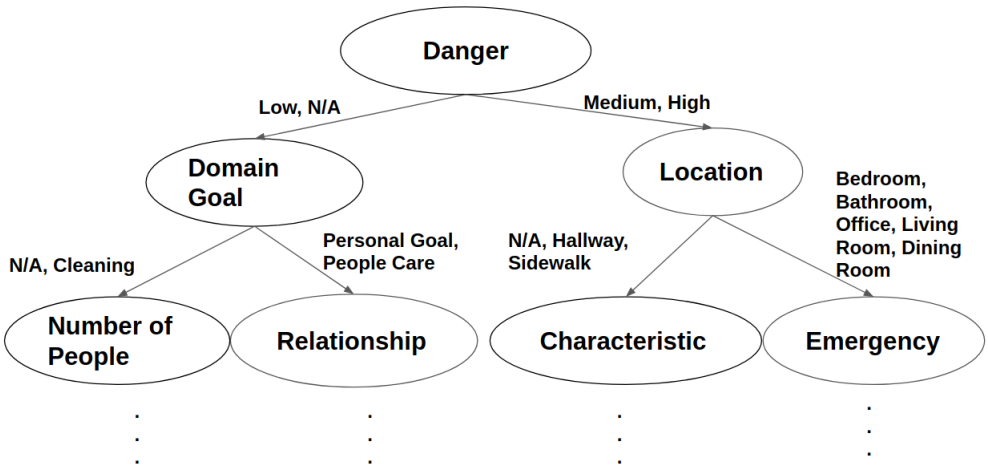



Figure 12: Decision Tree hierarchy illustration using the data from our human experiments with 11 different categorical context features.

of this regression tree contain the output priorities of each of the norms for the given context path, thereby adapting the precedence of norms depending on the situation.

### *7.4. Discussion*

The insight we can get from the above analysis is that the priority of norms is sensitive to the changing environment, yet in a complicated way.

First, the priority of norms are not globally static; instead, it is dependent on the context. Taking the cleaning task as an example, the robot might need to decide between finishing the task efficiently or carefully to ensure safety. In this scenario, participants reported significantly more preferences for the safety norm over the efficiency norm when there were people in the room. Here the priorities of two norms do not have a single ordering because they are mainly influenced by the feature, number of people. Such findings confirm our initial motivation of building a norm-aware reasoning framework, because autonomous systems have no way to solve norm conflicts without a correct perception of the context and the ability to reason based on it. Considering the complex nature of this problem, our proposed context representation and norm reasoning framework in section 6 are meaningful attempts towards a feasible solution for autonomous systems to deal with norm conflicts.

Some norms are especially more context-dependent than others. For example, the priority of safety norm is significantly influenced by five different context

features, but norms like 'do not harm' were not sensitive to any context variables in our settings. Our finding is that some norms are more universal while others are more context-specific, which be reflected by their context and activation conditions. By using the modular normative MDP we proposed in section 5, only relevant modules with a subset of appropriate norms are activated and computed in a given context which brings significant benefit in computation time and memory usage.

Context features play different roles in activating social norms; some of them are more influential than others. Based on the results of the ANOVAs, we can see that the priority of a norm can be sensitive to various contextual factors, yet the influence of each factor varies in terms of significance and effect size. As shown in our results, factors such as 'domain goal', 'emergency' and 'characteristics' dramatically influence the norms the robot should obey. Accordingly, those features should be given higher priorities or weights in robot perception of the environment to facilitate the normative reasoning process. The context tree we built based on human data in section 7.3 serves this purpose by starting checking with the most important features in the environment until reaching a reasonable norm output.

Additionally, the naturally occurring relations among contexts and entities in the world influence the features likely to occur and interact with each other. For example, particular characteristics like being a burglar might be naturally related to danger or emergency norms. The interaction effect of features on norm priorities increase the complexity of normative reasoning problem in addition. Considering the structure of human norm network, naive hard-coding rules are not enough for autonomous agents to socially interact with people in the dynamic context. To better capture the relationship, a more sophisticated modeling method is required to serve the computational framework, which refers to the pipeline we introduced in the section 4, 5, and 6.

## 8. Conclusion and Future Work

In this work, we proposed a scalable, and generalizable approach towards integrated norm-aware reasoning for autonomous systems. We proposed the MNMDP framework which provides a generalizable representation for norms and integrates it into the domain planning alleviating the curse of dimensionality problem encountered by previous approaches. Our MNMDP framework is well-suited for long-term autonomy applications since it needs little re-computation for any modifications made to the norm set. We illustrated its working using a roadway

simulator modeling traffic rules and also showed the computational advantages of using our MNMDP framework over existing methods. We motivated the need to model the activation function of the MNMDPs to successfully deploy the MNMDP approach in complex environments. We showed that our novel approach, which combines propositional logic and decision trees, for modeling the activation function is expressive, scalable, and interpretable. We illustrated the magnitude of the performance boost of our context model compared to a baseline with respect to both computation time and memory consumption. Using our context modeling approach, we demonstrated that we can efficiently identify which MNMDPs to use in various environmental conditions. Additionally, we showed that we can resolve normative conflicts using norm priorities, which are conditioned on the environmental context, rather than on global priorities as in previous work. Furthermore, we developed and deployed human experiments to model context-dependent norm priorities and to enable our context model to learn from the collected user data. These experiments also shed light on various crucial norm properties, such as dynamic norm priority, weighted contextual feature, context sensitivity and interaction effect between contextual factors.

As future work, we plan to make extensions to various parts of our approach. For the MNMDP framework, we aim to extend our approach to support partially observable environments. Additionally, we plan to learn estimates of the reward and penalty functions for the MNMDPs directly from user behavior using inverse reinforcement learning. Furthermore, we want to build a more concrete representation of the human normative reasoning process to improve our human norm network. To accomplish this goal, we plan to better capture the hierarchical structure and mutual activation of norms.

For the context modeling component, we plan to make the following extensions. First, we will extend our knowledge base with more diverse, yet realistic, social scenarios to improve the robustness of our models and enable our generalization approach to achieve better results. Second, we will extend our context tree to support online learning, which will make it better suited for long-term autonomy applications. To accomplish this, we will use an active learning component in which people in the environment can provide the robot with feedback regarding the appropriateness of its inferred activated norms. This feedback can then be used to create new connections in the knowledge graph and the context tree. Third, we plan to investigate the use of concepts from TWTL to extend our logic to support temporal relaxation for satisfying time constraints.

## 9. References


[1] G. Brennan, L. Eriksson, R. E. Goodin, N. Southwood, Explaining norms, Oxford University Press, 2013.

[2] M. S. Fagundes, S. Ossowski, J. Cerquides, P. Noriega, Design and evaluation of norm-aware agents based on normative markov decision processes, International Journal of Approximate Reasoning 78 (2016) 33–61.

[3] D. Kasenberg, M. Scheutz, Norm conflict resolution in stochastic domains, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[4] H. Aarts, A. Dijksterhuis, The silence of the library: environment, situational norm, and social behavior., Journal of personality and social psychology 84 (1) (2003) 18.

[5] V. Sarathy, M. Scheutz, Y. N. Kenett, M. Allaham, J. L. Austerweil, B. F. Malle, Mental representations and computational modeling of context-specific human norm systems, in: CogSci, 2017.

[6] J. Broersen, M. Dastani, J. Hulstijn, L. van der Torre, Goal generation in the BOID architecture, Cognitive Science Quarterly 2 (3-4) (2002) 428–447.

[7] M. J. Kollingbaum, T. J. Norman, Noa-a normative agent architecture, in: International Joint Conference on Artificial Intelligence, 2003, pp. 1465–1466.

[8] M. J. Kollingbaum, T. J. Norman, Norm adoption in the noa agent architecture, in: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, ACM, 2003, pp. 1038–1039.

[9] S. Panagiotidi, J. Vázquez-Salceda, Norm-aware planning: Semantics and implementation, in: IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2011, pp. 33–36.

[10] F. Meneguzzi, O. Rodrigues, N. Oren, W. W. Vasconcelos, M. Luck, Bdi reasoning with normative considerations, Engineering Applications of Artificial Intelligence 43 (2015) 127–146.

[11] N. Oren, W. Vasconcelos, F. Meneguzzi, M. Luck, Acting on norm constrained plans, in: International Workshop on Computational Logic in Multi-Agent Systems, Springer, 2011, pp. 347–363.

[12] M. S. Fagundes, S. Ossowski, M. Luck, S. Miles, Using normative markov decision processes for evaluating electronic contracts, AI Communications 25 (1) (2012) 1–17.

[13] J. Oh, F. Meneguzzi, K. Sycara, T. J. Norman, Prognostic normative reasoning, Engineering Applications of Artificial Intelligence 26 (2) (2013) 863–872.

[14] M. Brown, S. Saisubramanian, P. Varakantham, M. Tambe, Streets: game-theoretic traffic patrolling with exploration and exploitation, in: Twenty-Sixth IAAI Conference, 2014.

[15] W. W. Vasconcelos, M. J. Kollingbaum, T. J. Norman, Normative conflict resolution in multi-agent systems, Autonomous agents and multi-agent systems 19 (2) (2009) 124–152.

[16] J. S. Santos, J. O. Zahn, E. A. Silvestre, V. T. Silva, W. W. Vasconcelos, Detection and resolution of normative conflicts in multi-agent systems: a literature survey, Autonomous agents and multi-agent systems 31 (6).

[17] V. Krishnamoorthy, W. Luo, M. Lewis, K. Sycara, A computational framework for integrating task planning and norm aware reasoning for social robots, in: IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2018.

[18] V. Sarathy, M. Scheutz, B. Malle, Learning behavioral norms in uncertain and changing contexts, in: 8th IEEE International Conference on Cognitive Infocommunications, 2017.

[19] D. Kasenberg, M. Scheutz, Inverse norm conflict resolution, in: Proc. 1st AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society, 2018.

[20] G. Jeh, J. Widom, Simrank: A measure of structural-context similarity, in: Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[21] J. P. Gibbs, Norms: The problem of definition and classification, American Journal of Sociology 70 (5) (1965) 586–594.

[22] G. H. Von Wright, An essay in deontic logic and the general theory of action.

[23] H. Li, S. Milani, V. Krishnamoorthy, M. Lewis, K. Sycara, Perceptions of domestic robots' normative behavior across cultures, in: Proc. 2nd AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society, 2019.

[24] A. Pnueli, The temporal logic of programs, in: Proc. 18th Annual Symposium on Foundations of Computer Science, 1977.

[25] C.-I. Vasile, D. Aksaray, C. Belta, Time window temporal logic, in: Theoretical Computer Science, Vol. 691, 2017.

[26] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, Classification and regression trees. belmont, ca: Wadsworth, International Group (1984) 432.

[27] D. Coppersmith, S. J. Hong, J. R. Hosking, Partitioning nominal attributes in decision trees, Data Mining and Knowledge Discovery 3 (2) (1999) 197–217.

[28] G. De'Ath, Multivariate regression trees: a new technique for modeling species–environment relationships, Ecology 83 (4) (2002) 1105–1117.

[29] M. Johnson, K. Hofmann, T. Hutton, D. Bignell, The malmo platform for artificial intelligence experimentation, in: Proc. 25th International Joint Conference on Artificial Intelligence, 2016.

[30] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[31] I. F. of Robotics, World robotics industrial robots and service robots, Tech. rep. (2018).

[32] M. Scopelliti, M. V. Giuliani, F. Fornara, Robots in a domestic setting: a psychological approach, Universal access in the information society 4 (2) (2005) 146–155.

[33] J. E. Young, R. Hawkins, E. Sharlin, T. Igarashi, Toward acceptable domestic robots: Applying insights from social psychology, International Journal of Social Robotics 1 (1) (2009) 95.

[34] K. L. Koay, D. S. Syrdal, M. Ashgari-Oskoei, M. L. Walters, K. Dautenhahn, Social roles and baseline proxemic preferences for a domestic service robot, International Journal of Social Robotics 6 (4) (2014) 469–488.

[35] C.-A. Smarr, T. L. Mitzner, J. M. Beer, A. Prakash, T. L. Chen, C. C. Kemp, W. A. Rogers, Domestic robots for older adults: attitudes, preferences, and potential, International journal of social robotics 6 (2) (2014) 229–247.

[36] M. Pino, M. Boulay, F. Jouen, A. S. Rigaud, "are we ready for robots that care for us?" attitudes and opinions of older adults toward socially assistive robots, Frontiers in aging neuroscience 7 (2015) 141.

[37] W. Mason, S. Suri, Conducting behavioral research on amazon's mechanical turk, Behavior research methods 44 (1) (2012) 1–23.