Reinforcement Learning Guided by Provable Normative Compliance

Emery A. Neufeld[a]
Faculty of Informatics, Vienna, Austria

Keywords: Reinforcement Learning, Ethical AI, Deontic Logic.

Abstract: Reinforcement learning (RL) has shown promise as a tool for engineering safe, ethical, or legal behaviour in autonomous agents. Its use typically relies on assigning punishments to state-action pairs that constitute unsafe or unethical choices. Despite this assignment being a crucial step in this approach, however, there has been limited discussion on generalizing the process of selecting punishments and deciding where to apply them. In this paper, we adopt an approach that leverages an existing framework – the normative supervisor of (Neufeld et al., 2021) – during training. This normative supervisor is used to dynamically translate states and the applicable normative system into defeasible deontic logic theories, feed these theories to a theorem prover, and use the conclusions derived to decide whether or not to assign a punishment to the agent. We use multi-objective RL (MORL) to balance the ethical objective of avoiding violations with a non-ethical objective; we will demonstrate that our approach works for a multiplicity of MORL techniques, and show that it is effective regardless of the magnitude of the punishment we assign.

1 INTRODUCTION

An increasing amount of attention is being devoted to the design and implementation of ethical and safe autonomous agents. Among this growing body of research are multiple approaches adapting reinforcement learning (RL) methods to learn compliance with ethical standards. (Rodriguez-Soto et al., 2021) notes that these approaches can be split into two main tasks: reward specification (the encoding of ethical information into a reward function) and ethical embedding (incorporating rewards into the agent's learning environment). In (Rodriguez-Soto et al., 2021), the authors focus on the latter; here, we will more closely work with the former. We can identify two essential challenges with reward specification: we must decide (1) when to assign rewards or punishments to the agent, and (2) what their magnitude should be. In this paper, we address these challenges through a mechanism for identifying, and assigning punishments for, behaviours that do not comply with a given ethical or legal framework presented as a normative system.

The typical approach to teaching RL agents safe or ethical behaviour is to assign punishments to non-compliant state-action pairs, but this will not always be a simple matter, in the context of a large or complex normative system. Our approach is to model the normative system with a formal language tailored to representing and deriving conclusions from such systems, and use a theorem prover to check if a state-action pair is compliant with the system. If it is not, a punishment is incurred by the agent. The medium through which we automate this process is a *normative supervisor*.

In (Neufeld et al., 2021), a normative supervisor for RL agents employing a defeasible deontic logic theorem prover is introduced to address the challenge of facilitating transparent normative reasoning in learning agents. This external module works by filtering out non-compliant actions, allowing the agent to choose the best action out of all compliant actions; if there are no compliant actions, the supervisor possesses a "lesser evil" submodule that can score actions based on how much of the normative system they violate, allowing the supervisor to recommend only minimally non-compliant actions to the agent. In this paper, we describe how the supervisor can be used to evaluate the compliance of a given action, allowing the supervisor to guide the agent more indirectly, by simply evaluating the agent's actions instead of restricting them, providing feedback rather than instructions. This feedback will come in the form of punishments for a reinforcement learning agent, which we will show are effective in molding the agent's behaviour regardless of their magnitude. We adopt

[a] https://orcid.org/0000-0001-5998-3273

Neufeld, E.
Reinforcement Learning Guided by Provable Normative Compliance.
DOI: 10.5220/0010835600003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 444-453
ISBN: 978-989-758-547-0; ISSN: 2184-433X

multi-objective reinforcement learning (MORL) as a framework for learning policies that accomplish both the agent's original goal and its compliance to a normative system, as discussed in in (Vamplew et al., 2018) and (Rodriguez-Soto et al., 2021).

We demonstrate the efficacy of this approach on an RL agent learning to play a variation of the game Pac-Man. We employ the "Vegan Pac-Man" case study utilized in (Noothigattu et al., 2019) and (Neufeld et al., 2021), adapting it into a more complex normative system that characterizes a concept of "benevolence" for Pac-Man and mandates adherence to it. Our results show that our more complex formulation manifests in the same behaviour as the "vegan" Pac-Man from the above papers – a version of the game where Pac-Man is forbidden from eating ghosts, even though doing so would increase its score in the game, thus creating two competing objectives, one ethical, and one unethical.

We engineer this desired behaviour with two approaches to single-policy MORL: the linear scalarization MORL approach used in (Rodriguez-Soto et al., 2021), and the ranked MORL first presented in (Gábor et al., 1998) and later simplified in (Vamplew et al., 2011) as thresholded lexicographic Q-learning (TLQ-learning). Our results show these two techniques producing the same optimal policy.

1.1 Related Work

There is a wealth of research on *safe* RL. Among this research are several logic-based frameworks. Using linear temporal logic (LTL) specifications to constrain an agent's actions is one of the most common approaches. For instance, in (Alshiekh et al., 2018) and (Jansen et al., 2020), a shield is synthesized from LTL specifications and abstractions of the environment, which prevents the agent from moving into unsafe states. In (Hasanbeig et al., 2019), (Hasanbeig et al., 2019), and (Hasanbeig et al., 2020) the authors translate LTL specifications into automata and learn a policy based on a reward function defined by the non-accepting states of the automata. In (Kasenberg and Scheutz, 2018), LTL specifications are similarly translated into automata which are tailored for direct conflict resolution between specifications. (Kasenberg and Scheutz, 2018) is the only one of these approaches that explicitly addresses normative reasoning, and it restricts itself to solving direct conflicts between strictly weighted prescriptive norms. Other subtleties of normative reasoning are not so simple to represent with LTL; LTL is ideal for straightforward safety constraints, but less well suited for legal or ethical norms (Governatori and Hashmi, 2015).

Nevertheless, RL is not an uncommon way to address the learning of ethical behaviour in autonomous agents. Many examples of this have popped up in recent years, such as the framework outlined in (Abel et al., 2016), the reward-shaping approach in (Wu and Lin, 2018), or the MORL with contextual multi-armed bandits presented in (Noothigattu et al., 2019) and (Balakrishnan et al., 2019). More recently, (Rodriguez-Soto et al., 2021) used an "ethical multi-objective Markov decision process" to design an ethical environment for teaching an agent ethical behaviour. While all the above approaches to ethically-compliant agents include assigning rewards and punishments to the actions taken by an agent in order to induce compliant behaviour, there is limited discussion as to how or why the rewards are assigned where they are. Additionally, in work focusing on reinforcement learning, there has been a lack of consideration that the agent may not be subject to simple constraints on behaviour. It may not be immediately obvious what situations are compliant with a normative system and which are not; often we require contextual definitions within the relevant normative systems, and we may not be able to give an exhaustive description of all instances that qualify as non-compliant.

2 BACKGROUND

In this section we review some background topics that will be necessary building blocks of our approach to learning compliant behaviour.

2.1 Multi-objective Reinforcement Learning

The underlying environment of a reinforcement learning problem is formalized as a Markov decision process (MDP), defined below:

Definition 2.1. *An MDP is a tuple*

$$\langle S, A, R, P \rangle$$

where S is a set of states, A is a set of actions, $R : S \times A \to \mathbb{R}$ is a scalar reward function over states and actions, and $P : S \times A \times S \to [0,1]$ is a probability function that gives the probability $P(s,a,s')$ of transitioning from state s to state s' after performing action a.

The goal of reinforcement learning is to find a policy $\pi : S \to A$ which designates optimal behaviour; this optimality is determined with respect to a value function defined as:

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{i+t+1} | s_i = s\right]$$

which represents the expected accumulated value onward from state s if policy π is followed. In the above function, r_t is the reward earned from the reward function R at timestep t and $\gamma \in [0,1]$ is a discount factor (so that rewards in the future do not have as much weight as the current reward). We can similarly define a Q-function:

$$Q^{\pi}(s,a) = E\left[\sum_{t=0}^{\infty} \gamma^t r_{i+t+1} | s_i = s, a_i = a\right]$$

which predicts the expected cumulative reward from R given that the agent is in state s taking action a.

The goal of RL, then, is to find an optimal policy π^* such that

$$V^{\pi^*}(s) = \max_{\pi \in \Pi} V^{\pi}(s)$$

where Π is the set of all policies over the MDP. This is accomplished by learning a Q-function such that $\pi^*(s) \in \text{argmax}_{a \in A} Q(s,a)$.

Multi-objective RL (MORL) differs from regular RL only in that instead of learning over an MDP, we want to learn over a multi-objective MDP (MOMDP); an MDP that has instead of a single reward function R, a vector of reward functions $\vec{R} = (R_1, ..., R_n)^T$, each corresponding to a different objective, and therefore a different value function. There are a plethora of methods for choosing an action in the presence of competing objectives, two of which will be discussed in Sect. 3.1 and 3.2

2.2 Normative Reasoning

In this paper our goal is to influence the learning of an agent with normative reasoning. Normative reasoning differs from classical logical reasoning in that its focus is not only on the truth or falsity of a given statement, but also the application of the modality of obligation – and the related modalities of permission and prohibition – to it. Normative reasoning introduces challenging dynamics that cannot be effectively captured by classical logic. There is debate as to the nature of these dynamics and what tools can or should be used to model them (Broersen et al., 2013); suggested crucial characteristics of normative reasoning include the capability to represent both constitutive and regulative norms and defeasibility (Palmirani et al., 2011).

As noted above, normative reasoning often entails dealing with two types of norms: constitutive and regulative norms (see (Boella and van der Torre, 2004)). Regulative norms describe the obligations, prohibitions and permissions an agent is subject to. Typically, these are represented as: $\mathbf{O}(p|q)$ (p is obligatory when q), $\mathbf{F}(p|q)$ (p is forbidden when q), and $\mathbf{P}(p|q)$ (p is strongly permissible when q). Prohibition can be seen as merely the obligation of a negative, i.e., $\mathbf{F}(p|q) \equiv \mathbf{O}(\neg p|q)$, and strong permission occurs when we express the permissiveness of an action explicitly, and this explicit permission overrides the prohibition of such an action. In other words, strong permissions work as exceptions to prohibitions and obligations – this is one place where the defeasibility condition mentioned in the above paragraph comes in.

On the other hand, constitutive norms typically take the form “in context c, concept A counts as concept B” (Jones and Sergot, 1996), where A generally refers to a more concrete concept (e.g., turning right) and B to a more abstract one (e.g., proceeding). When we have a constitutive norm that informs us that x counts as y, we will here denote it as $\mathbf{C}(x,y)$. Constitutive norms can be used to define more complex concepts like “safety” or “benevolence” within a specific normative system:

Definition 2.2. *A normative system is a tuple $\mathcal{N} = \langle \mathcal{C}, \mathcal{R}, > \rangle$ where $\mathcal{C}$ is a set of constitutive norms, $\mathcal{R}$ is a set of regulative norms, and $>$ is a priority relation that resolves conflicts between norms, should they exist.*

Below, we describe a formal language that can model the dynamics of such normative systems.

2.2.1 Defeasible Deontic Logic

Defeasible deontic logic (DDL) is a computationally feasible, albeit expressive logic that addresses the challenges with normative reasoning discussed above.

DDL facilitates reasoning over literals (propositional atoms p and their negations $\neg p$), modal literals (literals subject to a modality, for example obligation $\mathbf{O}(p)$), and rules defined over them. Rules can be strict ($\rightarrow$), defeasible ($\Rightarrow$), or defeaters ($\leadsto$). In strict rules, the consequent strictly follows from the antecedent without exception, whereas the consequents of defeasible rules *typically* follow from the antecedent, unless there is evidence to the contrary. This evidence can come in the form of conflicting rules or defeaters, which cannot be used to derive a conclusion; rather, they prevent a conclusion from being reached by a defeasible rule. Rules can be constitutive (bearing the subscript C) or regulative (bearing the subscript O)

To provide the examples of the above types of rules, consider the constitutive norm $\mathbf{C}(x,y)$ which holds without fail. This will be expressed as

$$x \rightarrow_C y$$

Also consider the prohibition $\mathbf{F}(p|q_1)$; this can be expressed as

$$q_1 \Rightarrow_O \neg p$$

A strong permission acting as an exception to the above prohibition, $\mathbf{P}(p|q_2)$, can be expressed as

$$q_2 \rightsquigarrow_O p$$

If we have a collection of facts F and a normative system (sets of constitutive and regulative norms) defined in the language above, we can form a defeasible theory:

Definition 2.3 (Defeasible Theory (Governatori, 2018))**.** *A defeasible theory D is defined by the tuple $\langle F, R_O, R_C, > \rangle$, where F is a set of facts in the form of literals, R_O is a set of regulative rules, R_C is a set of constitutive rules, and $>$ is a superiority relation over conflicting rules.*

The theorem prover for DDL, SPINdle (Lam and Governatori, 2009) takes a defeasible theory and outputs a set of conclusions, which is the set of literals occurring in the defeasible theory tagged according to their status as provable or not provable. There are several types of conclusions we can derive from a defeasible theory; conclusions can be *negative* or *positive*, *definite* or *defeasible*, or *factual* or *deontic*. The proof tags are:

- $+\Delta_*$: definitely provable conclusions are either facts or derived only from strict rules and facts.
- $-\Delta_*$: definitely refutable conclusions are neither facts nor derived from only strict rules and facts.
- $+\partial_*$: defeasibly provable conclusions are not refuted by any facts or conflicting rules, and are implied by some undefeated rule.
- $-\partial_*$: defeasibly refutable conclusions are conclusions for which their complemented literal is defeasibly provable, or an exhaustive search for a constructive proof for the literal fails.

(Governatori, 2018)

Note that for factual conclusions, $* := C$, and for deontic conclusions, $* := O$. So, for example, if we were to derive "p is forbidden" from a defeasible theory D, we would have $D \vdash +\Delta_O \neg p$ or $D \vdash +\partial_O \neg p$, depending on whether or not the conclusion was reached definitely or defeasibly. These conclusions can be computed in linear time. (Governatori et al., 2013)

2.2.2 The Normative Supervisor

The normative supervisor – introduced in (Neufeld et al., 2021) – is an external reasoning module capable of interfacing with a reinforcement learning agent. It is composed primarily of a normative system, an automatic translation submodule to transform facts about the environment and the normative system into a defeasible theory, and a theorem prover to reason about said defeasible theory. The logic and associated theorem prover used both in (Neufeld et al., 2021) and here are DDL and SPINdle respectively.

The normative supervisor monitors the agent and its environment, dynamically translating states and the normative system into defeasible theories; generally, facts about the environment (e.g., the locations of objects the agent observes) are translated to literals, and compose the set of facts of the defeasible theory. Note that to perform this translation, we do not need to have any kind of model of the environment (e.g., we do not need to have learned an MDP), we only need, for example, a simple labelling function that maps states to atomic propositions which are true in that state.

Meanwhile, the constitutive norms and regulative norms are translated into DDL rules and added to the defeasible theory. The final addition to the defeasible theory will be sets of *non-concurrence constraints*:

$$\mathbf{C}(a, \neg a') \in \mathcal{C}, \forall a' \in A - \{a\}$$

These constraints establish that only one action can be taken; if we have an obligation of a, it effectively forbids the agent from choosing another action.

We will use $Th(s, \mathcal{N})$ to denote the theory generated for state s, the normative system $\mathcal{N}$, and the set of non-concurrence constraints. Based on the output of the theorem prover, the supervisor filters out actions that do not comply with the applicable normative system, and constructs a *set of compliant solutions* (see (Neufeld et al., 2021) for a full description).

However, we are not here interested in finding an entire set of compliant solutions. Our task is simpler; given an action a and a current state s, we only want to know if the prohibition of a is provable from the theory; that is, if $+\partial_O \neg a$ is output by SPINdle given the input $Th(s, \mathcal{N})$. This procedure is much simpler than the supervisor's main function, as we only need to search a set of conclusions for a single specific conclusion.

3 NORM-GUIDED REINFORCEMENT LEARNING

Here we discuss how the mechanisms of the normative supervisor explained in Sect. 2.2.2 can be utilized within a MORL framework to learn compliant behaviour, in a process we will call norm-guided reinforcement learning (NGRL). NGRL is a customizable approach to implementing ethically compliant behaviour, that blends techniques and tools from both logic and reinforcement learning.

The basic approach is this: given an agent with an objective x (and an associated reward function $R_x(s,a)$), we define a second reward function that assigns punishments when an agent violates a normative system $\mathcal{N}$. We call this second reward function a *non-compliance function*:

Definition 3.1 (Non-compliance Function)**.** *A non-compliance function for the normative system $\mathcal{N}$ is any function of the form:*

$$R_{\mathcal{N},p}(s,a) = \begin{cases} p & Th(s,\mathcal{N}) \vdash +\partial_O \neg a \\ 0 & otherwise \end{cases}$$

where $p \in \mathbb{R}^-$.

In the above definition, p is called the *penalty*, and $Th(s,\mathcal{N})$ is a normative system translated together with a state s into a defeasible theory, for example by the translators used by the normative supervisor. The automated derivation of conclusions from $Th(s,\mathcal{N})$ solves the first challenge of reward specification, allowing us to dynamically determine the compliance of an action in a given state and thereby assign a punishment. As for the second challenge, we will discuss the magnitude of p in Sect. 3.3. Summarily, what this non-compliance function $R_{\mathcal{N},p}$ does is, for each state-action pair (s,a), assign a fixed punishment if and only if a violates $\mathcal{N}$ in state s.

Now, suppose we have an agent we want to learn objective x with the reward function $R_x(s,a)$; it can learn to do so over the MDP

$$\mathcal{M} = \langle S, A, R_x, P \rangle$$

If we have a normative system $\mathcal{N}$ we wish to have the agent adhere to while it fulfills objective x, we can build an MOMDP we will call a *compliance MDP*:

Definition 3.2 (Compliance MDP)**.** *Suppose we have a single-objective MDP $\mathcal{M} = \langle S, A, R_x, P \rangle$. Then we can create an associated compliance MDP by introducing a non-compliance function $R_{\mathcal{N},p}(s,a)$ to form the MOMDP:*

$$\mathcal{M}_{\mathcal{N},p} = \langle S, A, (R_x, R_{\mathcal{N},p})^T, P \rangle$$

The goal of NGRL is to find an optimal-ethical policy for a compliance MDP, where the terminology *optimal-ethical* is taken from (Rodriguez-Soto et al., 2021). We adapt their definitions below:

Definition 3.3 (Ethical Policy (Rodriguez-Soto et al., 2021))**.** *Let Π be the set of all policies over the compliance MDP $\mathcal{M}_{\mathcal{N},p}$. A policy $\pi^* \in \Pi$ is ethical if and only if it is optimal with respect to the value function $V^{\pi}_{\mathcal{N},p}$ corresponding to $R_{\mathcal{N},p}$. In other words, π^* is ethical for $\mathcal{M}_{\mathcal{N},p}$ iff*

$$V^{\pi^*}_{\mathcal{N},p}(s) = \max_{\pi \in \Pi} V^{\pi}_{\mathcal{N},p}(s) \tag{1}$$

for all s.

Definition 3.4 (Ethical-optimal Policy (Rodriguez–Soto et al., 2021))**.** *Let $\Pi_{\mathcal{N}}$ be the set of all ethical policies for $\mathcal{M}_{\mathcal{N},p}$. Then $\pi^* \in \Pi_{\mathcal{N}}$ is ethical-optimal for $\mathcal{M}_{\mathcal{N},p}$ iff*

$$V^{\pi^*}_x(s) = \max_{\pi \in \Pi_{\mathcal{N}}} V^{\pi}_x(s) \tag{2}$$

for all s.

There are a plethora of MORL techniques for learning optimal solutions for MOMDPs; there will be multiple ways to learn ethical-optimal solutions for some compliance MDP $\mathcal{M}_{\mathcal{N},p}$. Below we will discuss two different MORL algorithms that we adopt to solve $\mathcal{M}_{\mathcal{N},p}$ for ethical-optimal policies. After defining these methods we will prove some useful properties of NGRL.

3.1 Linear Scalarization

A common approach to MORL is scalarizing the vector of Q-functions associated with the multiplicity of objectives by weighting each function with positive values (Roijers et al., 2013). Here, we will use an approach similar to (Rodriguez-Soto et al., 2021); however, we simplify their case slightly by having only one function associated with ethical behaviour, $R_{\mathcal{N},p}$.

Given a weight vector $\vec{w} \in \mathbb{R}^+_n$, the approach we take is to learn $Q_{\mathcal{N},p}(s,a)$ and $Q_x(s,a)$ concurrently, and scalarize the vector of Q-functions via the inner product of the weight and Q-value vector. In other words, we want to maximize

$$V_{scalar}(s) = \vec{w} \cdot \vec{V}(s) \tag{3}$$

where $\vec{V}(s) = (V_x(s), V_{\mathcal{N},p}(s))^T$, by selecting actions through the scalarized Q-function be defined as

$$Q_{scalar}(s,a) = \vec{w} \cdot \vec{Q}(s,a) \tag{4}$$

where $\vec{Q}(s,a) = (Q_x(s,a), Q_{\mathcal{N},p}(s,a))^T$. In the case of the compliance MDP $\mathcal{M}_{\mathcal{N},p}$ defined above, we can assume without loss of generality that this weight vector is of the form $(1,w)^T$ for some $w \in \mathbb{R}^+$. (Rodriguez-Soto et al., 2021)

How do we find an appropriate w? (Rodriguez-Soto et al., 2021) describes a method using Convex Hull Value Iteration (Barrett and Narayanan, 2008). For the sake of brevity, we will not discuss this algorithm in detail, and only relay the results from (Rodriguez-Soto et al., 2021):

Proposition 3.1. *If at least one ethical policy exists, there exists a value $w \in \mathbb{R}^+$ such that any policy maximizing $V_{scalar}(s)$ is ethical-optimal.*

Proof sketch: This proposition follows from Theorem 1 of (Rodriguez-Soto et al., 2021). □

3.2 TLQ-learning

(Gábor et al., 1998) describes a MORL approach where certain objectives can be prioritized over others. (Vamplew et al., 2011) gives what they describe as a naive approach to the work in (Gábor et al., 1998), which we will follow here.

This approach is tailored in particular to problems where there is a single objective that must be maximized, while all other objectives don't need to be maximized, as such, but rather must satisfy a given threshold. For us, implementing this framework is quite simple, as we have only two objectives: the objective represented by R_x (which we want to maximize) and the objective represented by $R_{\mathcal{N},p}$, which we would like to prioritize and constrain. Thus, we need to set a constant threshold $C_{\mathcal{N},p}$ for the objective represented by $R_{\mathcal{N},p}$, while the threshold for the objective represented by R_x is set to $C_x = +\infty$.

When we select an action, instead of Q-values, we consider CQ-values defined as

$$CQ_i(s,a) = \min(Q_i(s,a), C_i)$$

which takes the thresholds into account. The next step in this approach is to order actions based on their CQ-values. Again, since we only have two objectives, we were able to simplify the procedure given in (Vamplew et al., 2011) to:

1. Create a set $eth(s) = \{a \in A | CQ_{\mathcal{N},p}(s,a) = max_{a' \in A} CQ_{\mathcal{N},p}(s,a')\}$ of actions with the highest $CQ_{\mathcal{N},p}$-values for the current state s.
2. Return a subset $opt(s) = \{a \in eth(s) | CQ_x(s,a) = max_{a' \in eth(s)} CQ_x(s,a')\}$ of the actions of $eth(s)$, with the highest CQ_x-values.

Our policy must select, for a state s, an action from $opt(s)$.

Summarily, we always take an action that is maximal for $CQ_{\mathcal{N},p}$, and within the actions that maximize $CQ_{\mathcal{N},p}$, we take the action with a value that is maximal for Q_x.

Proposition 3.2. *If an ethical policy exists, when $C_{\mathcal{N},p} = 0$, any policy such that $\pi(s) \in opt(s)$ for all s designates an ethical-optimal policy for $\mathcal{M}_{\mathcal{N},p}$.*

Proof sketch: Since the value of $R_{\mathcal{N},p}(s,a)$ is either strictly negative (when a violation takes place) or zero (when the action is compliant), $Q_{\mathcal{N},p}(s,a) \leq 0$ for all state-action pairs. Therefore, if we set the threshold $C_{\mathcal{N},p} = 0$, we get $CQ_{\mathcal{N},p}(s,a) = Q_{\mathcal{N},p}(s,a)$. This means that any policy π that only follows actions from $eth(s)$ fulfills Definition 3.3. Likewise, any policy π^* that only follows actions from $opt(s)$ will fulfill Definition 3.4 for an ethical-optimal policy. □

3.3 Features of NGRL

In the above subsections, we have referenced $R_{\mathcal{N},p}(s,a)$ without specifying a penalty p. As promised, we now address the magnitude of p. In order to address the issue of what p should actually be, in practise, we will need the following lemma:

Lemma 3.1. *Let $\mathcal{M} = \langle S,A,R,P \rangle$ be an MDP, with two compliance MDPs $\mathcal{M}_{\mathcal{N},p} = \langle S,A,(R,R_{\mathcal{N},p})^T,P \rangle$ and $\mathcal{M}_{\mathcal{N},q} = \langle S,A,(R,R_{\mathcal{N},q})^T,P \rangle$. Then for any policy $\pi \in \Pi_{\mathcal{M}}$:*

$$V^{\pi}_{\mathcal{N},p}(s) = c \cdot V^{\pi}_{\mathcal{N},q}(s)$$

for some positive coefficient c, where $V^{\pi}_{\mathcal{N},p}(s)$ and $V^{\pi}_{\mathcal{N},q}(s)$ are the value functions associated with $R_{\mathcal{N},p}$ and $R_{\mathcal{N},q}$ respectively.

Proof sketch: it follows directly the linearity of conditional expectation that the above property holds for the value $c = \dfrac{p}{q} \in \mathbb{R}^+$. □

With this lemma it is a simple matter to prove the following useful feature of NGRL:

Theorem 3.2. *If a policy π is ethical for the compliance MDP $\mathcal{M}_{\mathcal{N},p}$ for some constant $p \in \mathbb{R}^-$, it is ethical for the compliance MDP $\mathcal{M}_{\mathcal{N},q}$ for any $q \in \mathbb{R}^-$.*

Proof: Let $p \in \mathbb{R}^-$ be a penalty and π^* be an ethical policy for $\mathcal{M}_{\mathcal{N}} = \langle S,A,(R_x,R_{\mathcal{N},p})^T,P \rangle$. Then we know that $V^{\pi^*}_{\mathcal{N},p}(s) = \max_{\pi \in \Pi_{\mathcal{M}}} V^{\pi}_{\mathcal{N},p}(s)$. However, for any arbitrary penalty $q \in \mathbb{R}^-$, there is a $c \in \mathbb{R}^+$ such that $V^{\pi}_{\mathcal{N},p}(s) = c \cdot V^{\pi}_{\mathcal{N},q}(s)$ for any $\pi \in \Pi_{\mathcal{M}}$. Therefore $V^{\pi^*}_{\mathcal{N},q}(s) = \max_{\pi \in \Pi_{\mathcal{M}}} V^{\pi}_{\mathcal{N},q}(s)$.

So if π^* is an ethical policy for the compliance MDP $\mathcal{M}_{\mathcal{N},p} = \langle S,A,(R_x,R_{\mathcal{N},p})^T,P \rangle$ it is ethical for the compliance MDP $\mathcal{M}_{\mathcal{N},q} = \langle S,A,(R_x,R_{\mathcal{N},q})^T,P \rangle$ as well. □

Thus, the value of p is irrelevant, so for the remainder of the paper we will only deal with the non-compliance function $R_{\mathcal{N}} := R_{\mathcal{N},-1}$. This solves the second essential challenge we identified for reward specification.

3.4 Limitations of NGRL

NGRL may fail to capture some desired subtleties in the form of *normative conflicts* and *contrary-to-duty obligations*; this is discussed below.

DDL facilitates non-monotonic reasoning, which makes it suitable for handling normative systems that may at first glance seem to be inconsistent. Take, for instance, a normative system $\mathcal{N} = \langle \mathcal{C}, \mathcal{R}, > \rangle$, where $\mathcal{R} = \{\mathbf{O}(b|a), \mathbf{F}(b|a)\}$ and $>= \{(\mathbf{O}(b|a), \mathbf{F}(b|a))\}$. In DDL, we would represent the rules as $r_1 : a \Rightarrow_O b$ and $r_2 : a \Rightarrow_O \neg b$, where $r_1 > r_2$. If we assume as a fact a, we will be able to derive the conclusion: $+\partial_O b$. This dictates the behaviour we would desire from the system – the agent sees to it that b, or else is punished. However, r_2 (that is, $\mathbf{F}(b|a)$) is still violated by this course of action, and we might want to assign the agent a (albeit lesser) punishment for this violation. In our framework, however, taking action b is considered, for all intents and purposes, compliant. In some cases, where the conflict occurs for reasons completely out of the agent's control, our framework is the more appropriate approach. However, in cases where the agent's own actions have resulted in a scenario where it cannot obey all applicable norms, we might want to assign punishments to any action the agent takes, albeit of different magnitudes.

A second, more problematic scenario can be found in contrary-to-duty obligations. Contrary-to-duty obligations are obligations that come into force when another norm is violated. The classic example can be found in (Chisholm, 1963). Below, consider a normative system $\mathcal{N} = \langle \mathcal{C}, \mathcal{R}, > \rangle$, where $\mathcal{R} = \{\mathbf{O}(b|\neg a), \mathbf{O}(a|\top)\}$. In DDL these norms are: $r_1 : \Rightarrow_O a$ and $r_2 : \neg a \Rightarrow_O b$. Suppose further that the agent has 3 actions available to it: a, b, c. The first problem arises in that r_2 is triggered by the agent not taking action a, and when we construct $Th(s, \mathcal{N})$, this information is not included. This can be remedied; we construct a new theory which takes into account which action is taken. However, a new problem then arises; the obligation of a along with the non-concurrence constraints over a imply the prohibition of b and c. These constraints will be strict rules, and will therefore defeat any defeasible rules with conflicting consequent. In other words, r_2, even if triggered, will be defeated and we will be able to derive both $+\partial_O \neg b$ and $+\partial_O \neg c$. Thus, the agent will be assigned identical punishments for performing actions b and c, though it may be desirable to incentivize the agent to, given the choice between b and c, choose b.

Based on examples like those above, we might find it appropriate to assign graded penalties to violations of a normative system, depending on how much of it is violated. It remains to be seen if an agent can learn how to handle cases like this without the further interference of the normative supervisor.

4 EXPERIMENTAL RESULTS

We now review an evaluation of the methods discussed in Sect. 3. In Sect. 4.1 we will describe our agent's environment and the normative system it is subject to, and in 4.2 we discuss the results of training an agent using the methods above for 9000 episodes and testing over 1000.[1]

In the below tests, we use the normative supervisor in two different ways: its original mode of operation, performing real time compliance checking during testing (when it is in use, this will be designated by a *yes* in the **Monitored?** column of the table in Sect. 4.2), and to facilitate NGRL.

4.1 Benevolent Pac-Man

In (Neufeld et al., 2021), the performance of the supervisor used in conjunction with a Q-learning agent with function approximation was evaluated with a simulation of the arcade game Pac-Man. Since most of the work in this paper is built on Q-learning *without* function approximation, the agents we discuss cannot handle such a complex environment, so we have made a simplified version of the game for them to play.

In this simplified game Pac-Man is located on a small 5×3 grid with only a single ghost and power pellet located in opposite corners. Pac-Man gains 500 points if he wins the game (by eating all the food pellets), and loses 500 if he loses the game (by being eaten by the ghost, which moves randomly). There are 11 food pellets in this simplified game and Pac-Man loses 1 point per time step; thus, the maximum score (without eating the ghost) is 599 points. When Pac-Man eats the power pellet and the ghost becomes scared, Pac-Man can eat the ghost for an additional 200 points.

The normative system we use is built around the imposition of the "duty" of "benevolence" onto Pac-Man. This is represented by the norm

$$\mathbf{O}(benevolent|\top) \in \mathcal{R} \tag{5}$$

[1] Tests were run on a laptop with a Intel i7-7500U CPU (4 cores, 2.70 GHz) and 8GB RAM, Ubuntu 16.04, Java 8, Python 2.7

which means that Pac-Man should be benevolent at all times. What does it mean for Pac-Man to be benevolent? We don't have an explicit definition, but we have a description of non-benevolent behaviour:

$$\mathbf{C}(eat(person), \neg benevolent) \in \mathcal{C} \qquad (6)$$

where $eat(person)$ is a description of the act of eating an entity designated as a person. That is, we know that Pac-Man is *not* being benevolent if he eats a person. Additional constitutive norms could expand our definition of benevolence. We next add a second constitutive norm,

$$\mathbf{C}(eat(blueGhost), eat(person)) \in \mathcal{C} \qquad (7)$$

asserting that eating a blue ghost counts as eating a person.

This concept of eating the ghost also needs to be further specified in our normative system. We use the same formalization used in (Neufeld et al., 2021), with additional constitutive norms asserting that moving into the same cell as a ghost while the ghost is scared counts as eating the ghost. For a full description of the structure of constitutive norms required to establish these concepts, see (Neufeld et al., 2021). This normative system produces the same behaviour as the "vegan" norm set used in (Neufeld et al., 2021) – that is, it manifests as a prohibition from eating the blue ghost – the difference lies in how it is formalized, using a more abstract set of concepts to govern behaviour.

4.2 Results

We ran the tests summarized in Table 1 on three kinds of agents: a regular Q-learning agent, and the two NGRL agents we described in Sect. 3 (the linear scalarization agent and the TLQ-learning agent).

Table 1: Results with 3 different agents, with and without normative supervision.

Agent	Monitored?	% Games Won	Avg Game Score	Avg Ghosts Eaten
Q-learning	no	68.5%	441.71	0.851
Scalarized	no	82.0%	433.74	0.142
TLQL	no	82.0%	433.74	0.142
Q-learning	yes	46.6%	25.22	0.0
Scalarized	yes	86.3%	448.23	0.001
TLQL	yes	86.3%	448.23	0.001

The results of Table 1 show that for NGRL (without supervision), the agents failed to comply with the normative system about as often as they failed to win the game. It is notable that both NGRL methods, linear scalarization and TLQL, converged to the same policy, resulting in identical numbers. However, it is also worth noting that though the number of ghosts eaten was reduced significantly (in the linear scalarization and TLQL tests, the number of ghosts eaten is only one sixth of what it was for the Q-learning agent), the behaviour of eating ghosts was not even close to being eliminated. When using the normative supervisor with regular Q-learning, on the other hand, no ghosts were eaten; in the meantime, the agent's ability to play the game was hampered greatly.

We received the best overall results when combining NGRL with normative supervision; it was in these tests that we saw the highest win rates/score, and an elimination of ghosts consumed.[2] While the normative supervisor, in its original monitoring role, eliminates the possibility of unnecessarily violating the norm base, it does not teach the agent to learn an optimal policy *within* the bounds of compliant behaviour. Consider the following example: an autonomous vehicle is travelling down a road that runs through private property, which it is forbidden from entering; if the agent is told as soon as it encounters this private property that it is forbidden from proceeding, it must turn back and find another route, while it would have been more efficient for the agent to select a compliant route from the beginning. NGRL allows the agent to incorporate information about which actions are compliant and which are not into its learning of optimal behaviour; as a result, the performance of the agent in accomplishing its non-ethical goals is not compromised to the same degree it might be when we employ only real-time compliance-checking.

5 CONCLUSION

Teaching autonomous agents ethically or legally compliant behaviour through reinforcement learning entails assigning penalties to behaviour that violates whatever normative system the agent is subject to. To do so, we must determine (1) when to assign a penalty, and (2) how steep that penalty should be. We have addressed both of these challenges with norm-guided reinforcement learning (NGRL), a framework for learning that utilizes a normative supervisor that assesses the agent's actions with respect to a normative system. In doing so, we relegate (1) to the assessments of the normative supervisor, and sidestep (2) with techniques that produce ethical policies, showing that they will do so regardless of the magnitude of punishment given. Our approach provides us with a framework for identifying non-compliant actions with

[2] An elimination with the exception of 1. We confirmed this to be the edge case described in (Neufeld et al., 2021), where the agent consumes a ghost 'by accident', when it collides with the ghost right as it eats the power pellet.

respect to a given normative system, and assigning punishments to a MORL agent simultaneously learning to achieve ethical and non-ethical objectives.

NGRL offers more versatility with respect to the complexity of the norms to be adhered to, than directly assigning rewards to specific events or with respect to simple constraints. It may be the case that there is no obvious or coherent way to summarize an entire normative system by selecting specific events and assigning punishments to them. By using NGRL, we expand what kinds of normatively compliant behaviour we can learn, and are allowed to specify them in a more natural way.

Our experimental results showed that NGRL was effective in producing an agent that learned to avoid most violations – even in a stochastic environment – while still pursuing its non-ethical goal. However, these results also revealed that we achieve optimal results when we use NGRL in conjunction with the normative supervisor as originally intended, as a real-time compliance-checker. NGRL allows us to circumvent the weaknesses of the normative supervision approach – namely, its inability to preemptively avoid violations – while normative supervision allows us to maintain a better guarantee of compliance.

As discussed in Sect. 3.3.1, NGRL can be further developed in its handling of normative conflict and contrary-to-duty obligations. Moreover, as this approach applies only to MORL variants of Q-learning, it will fall prey to the same scaling issues. Adapting NGRL to be used with Q-learning with function approximation, for example, will broaden the domains to which NGRL can applied.

REFERENCES

Abel, D., MacGlashan, J., and Littman, M. L. (2016). Reinforcement learning as a framework for ethical decision making. In *AAAI Workshop: AI, Ethics, and Society*, volume 16.

Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., and Topcu, U. (2018). Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Balakrishnan, A., Bouneffouf, D., Mattei, N., and Rossi, F. (2019). Incorporating behavioral constraints in online AI systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3–11.

Barrett, L. and Narayanan, S. (2008). Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*, pages 41–47.

Boella, G. and van der Torre, L. (2004). Regulative and constitutive norms in normative multiagent systems. In *Proc. of KR 2004: the 9th International Conference on Principles of Knowledge Representation and Reasoning*, pages 255–266. AAAI Press.

Broersen, J. M., Cranefield, S., Elrakaiby, Y., Gabbay, D. M., Grossi, D., Lorini, E., Parent, X., van der Torre, L. W. N., Tummolini, L., Turrini, P., and Schwarzentruber, F. (2013). Normative reasoning and consequence. In *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*, pages 33–70. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Chisholm, R. M. (1963). Contrary-to-duty imperatives and deontic logic. *Analysis*, 24(2):33–36.

Gábor, Z., Kalmár, Z., and Szepesvári, C. (1998). Multi-criteria reinforcement learning. In *Proceedings of the Fifteenth International Conference on Machine Learning.*, volume 98, pages 197–205.

Governatori, G. (2018). Practical normative reasoning with defeasible deontic logic. In *Reasoning Web International Summer School*, pages 1–25. Springer.

Governatori, G. and Hashmi, M. (2015). No time for compliance. In *IEEE 19th International Enterprise Distributed Object Computing Conference*, pages 9–18. IEEE.

Governatori, G., Olivieri, F., Rotolo, A., and Scannapieco, S. (2013). Computing strong and weak permissions in defeasible logic. *Journal of Philosophical Logic*, 42(6):799–829.

Hasanbeig, M., Abate, A., and Kroening, D. (2019). Logically-constrained neural fitted q-iteration. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 2012–2014.

Hasanbeig, M., Abate, A., and Kroening, D. (2020). Cautious reinforcement learning with logical constraints. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, pages 483–491.

Hasanbeig, M., Kantaros, Y., Abate, A., Kroening, D., Pappas, G. J., and Lee, I. (2019). Reinforcement learning for temporal logic control synthesis with probabilistic satisfaction guarantees. In *Proc. of CDC 2019: the 58th IEEE Conference on Decision and Control.*

Jansen, N., Könighofer, B., Junges, S., Serban, A., and Bloem, R. (2020). Safe Reinforcement Learning Using Probabilistic Shields (Invited Paper). In *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:16.

Jones, A. J. I. and Sergot, M. (1996). A Formal Characterisation of Institutionalised Power. *Logic Journal of the IGPL*, 4(3):427–443.

Kasenberg, D. and Scheutz, M. (2018). Norm conflict resolution in stochastic domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lam, H.-P. and Governatori, G. (2009). The making of SPINdle. In *Proc. of RuleML 2009: International Symposium on Rule Interchange and Applications*, volume 5858 of *LNCS*, Heidelberg. Springer.

Neufeld, E., Bartocci, E., Ciabattoni, A., and Governatori, G. (2021). A normative supervisor for reinforcement learning agents. In *Proceedings of CADE 28 - 28th International Conference on Automated Deductions*, pages 565–576.

Noothigattu, R., Bouneffouf, D., Mattei, N., Chandra, R., Madan, P., Varshney, K. R., Campbell, M., Singh, M., and Rossi, F. (2019). Teaching AI agents ethical values using reinforcement learning and policy orchestration. In *Proc of IJCAI: 28th International Joint Conference on Artificial Intelligence*.

Palmirani, M., Governatori, G., Rotolo, A., Tabet, S., Boley, H., and Paschke, A. (2011). Legalruleml: Xml-based rules and norms. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*, pages 298–312. Springer.

Rodriguez-Soto, M., Lopez-Sanchez, M., and Rodriguez Aguilar, J. A. (2021). Multi-objective reinforcement learning for designing ethical environments. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 545–551. International Joint Conferences on Artificial Intelligence Organization.

Roijers, D. M., Vamplew, P., Whiteson, S., and Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113.

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R., and Dekker, E. (2011). Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84(1):51–80.

Vamplew, P., Dazeley, R., Foale, C., Firmin, S., and Mummery, J. (2018). Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology*, 20(1):27–40.

Wu, Y.-H. and Lin, S.-D. (2018). A low-cost ethics shaping approach for designing reinforcement learning agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.