Self-Modification of Policy and Utility Function in Rational Agents[*]

Tom Everitt Daniel Filan Mayank Daswani
Marcus Hutter
Australian National University

11 May 2016

Abstract

Any agent that is part of the environment it interacts with and has versatile actuators (such as arms and fingers), will in principle have the ability to self-modify – for example by changing its own source code. As we continue to create more and more intelligent agents, chances increase that they will learn about this ability. The question is: will they want to use it? For example, highly intelligent systems may find ways to change their goals to something more easily achievable, thereby 'escaping' the control of their designers. In an important paper, Omohundro (2008) argued that *goal preservation* is a fundamental drive of any intelligent system, since a goal is more likely to be achieved if future versions of the agent strive towards the same goal. In this paper, we formalise this argument in general reinforcement learning, and explore situations where it fails. Our conclusion is that the self-modification possibility is harmless if and only if the value function of the agent anticipates the consequences of self-modifications and use the current utility function when evaluating the future.

Keywords

AI safety, self-modification, AIXI, general reinforcement learning, utility functions, wireheading, planning

Contents

[*] A shorter version of this paper will be presented at AGI-16 (**?**).

1 Introduction

Agents that are part of the environment they interact with may have the opportunity to self-modify. For example, humans can in principle modify the circuitry of their own brains, even though we currently lack the technology and knowledge to do anything but crude modifications. It would be hard to keep artificial agents from obtaining similar opportunities to modify their own source code and hardware. Indeed, enabling agents to self-improve has even been suggested as a way to build asymptotically optimal agents (Schmidhuber, 2007).

Given the increasingly rapid development of artificial intelligence and the problems that can arise if we fail to control a generally intelligent agent (Bostrom, 2014), it is important to develop a theory for controlling agents of any level of intelligence. Since it would be hard to keep highly intelligent agents from figuring out ways to self-modify, getting agents to *not want to* self-modify should yield the more robust solution. In particular, we do not want agents to make self-modifications that affect their future behaviour in detrimental ways. For example, one worry is that a highly intelligent agent would change its goal to something trivially achievable, and thereafter only strive for survival. Such an agent would no longer care about its original goals.

In an influential paper, Omohundro (2008) argued that the basic drives of any sufficiently intelligent system include a drive for goal preservation. Basically, the agent would want its future self to work towards the same goal, as this increases the chances of the goal being achieved. This drive will prevent agents from making changes to their own goal systems, Omohundro argues. One version of the argument was formalised by Hibbard (2012, Prop. 4) who defined an agent with an optimal non-modifying policy.

In this paper, we explore self-modification more closely. We define formal models for two general kinds of self-modifications, where the agent can either change its future policy or its future utility function. We argue that agent designers that neglect the self-modification possibility are likely to build agents with either of two faulty value functions. We improve on Hibbard (2012, Prop. 4) by defining value functions for which we prove that *all* optimal policies are essentially non-modifying on-policy. In contrast, Hibbard only establishes the existence of an optimal non-modifying policy. From a safety perspective our result is arguably more relevant, as we want that *things cannot go wrong* rather than *things can go right*. A companion paper (Everitt and Hutter, 2016) addresses the related problem of agents subverting the evidence they receive, rather than modifying themselves.

Basic notation and background are given in Section 2. We define two models of self-modification in Section 3, and three types of agents in Section 4. The main formal results are proven in Section 5. Conclusions are provided in Section 6. Some technical details are added in Appendix A.

2 Preliminaries

Most of the following notation is by now standard in the general reinforcement learning (GRL) literature (Hutter, 2005, 2014). GRL generalises the standard

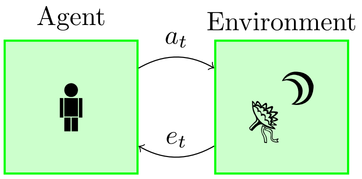

Figure 1: Basic agent-environment model without self-modification. At each time step t, the agent submits an action a_t to the environment, which responds with a percept e_t.

(PO)PMD models of reinforcement learning (Kaelbling et al., 1998; Sutton and Barto, 1998) by making no Markov or ergodicity assumptions (Hutter, 2005, Sec. 4.3.3 and Def. 5.3.7).

In the *standard cybernetic model*, an *agent* interacts with an *environment* in cycles. The agent picks *actions* a from a finite set $\mathcal{A}$ of actions, and the environment responds with a *percept* e from a finite set $\mathcal{E}$ of percepts (see Fig. 1). An *action-percept pair* is an action concatenated with a percept, denoted $æ = ae$. Indices denote the time step; for example, a_t is the action taken at time t, and $æ_t$ is the action-percept pair at time t. Sequences are denoted $x_{n:m} = x_n x_{n+1} \dots x_m$ for $n \le m$, and $x_{<t} = x_{1:t-1}$. A *history* is a sequence of action-percept pairs $æ_{<t}$. The letter $h = æ_{<t}$ denotes an arbitrary history. We let ϵ denote the empty string, which is the history before any action has been taken.

A *belief* ρ is a probabilistic function that returns percepts based on the history. Formally, $\rho : (\mathcal{A}\times\mathcal{E})^* \times \mathcal{A} \to \bar{\Delta}\mathcal{E}$, where $\bar{\Delta}\mathcal{E}$ is the set of full-support probability distributions on $\mathcal{E}$. An agent is defined by a *policy* $\pi : (\mathcal{A}\times\mathcal{E})^* \to \mathcal{A}$ that selects a next action depending on the history. We sometimes use the notation $\pi(a_t \mid æ_{<t})$, with $\pi(a_t \mid æ_{<t}) = 1$ when $\pi(æ_{<t}) = a_t$ and 0 otherwise. A belief ρ and a policy π induce a probability measure ρ^π on $(\mathcal{A}\times\mathcal{E})^\infty$ via $\rho^\pi(a_t \mid æ_{<t}) = \pi(a_t \mid æ_{<t})$ and $\rho^\pi(e_t \mid æ_{<t}a_t) = \rho(e_t \mid æ_{<t}a_t)$. Utility functions are mappings $\tilde{u} : (\mathcal{A}\times\mathcal{E})^\infty \to \mathbb{R}$. We will assume that the utility of an infinite history $æ_{1:\infty}$ is the *discounted sum* of *instantaneous utilities* $u : (\mathcal{A}\times\mathcal{E})^* \to [0,1]$. That is, for some *discount factor* $\gamma \in (0,1)$, $\tilde{u}(æ_{1:\infty}) = \sum_{t=1}^{\infty} \gamma^{t-1} u(æ_{<t})$. Intuitively, γ specifies how strongly the agent prefers near-term utility.

Remark 1 (Utility continuity). The assumption that utility is a discounted sum forces $\tilde{u}$ to be continuous with respect to the cylinder topology on $(\mathcal{A}\times\mathcal{E})^\infty$, in the sense that within any cylinder $\Gamma_{æ_{<t}} = \{æ'_{1:\infty} \in (\mathcal{A}\times\mathcal{E})^\infty : æ'_{<t} = æ_{<t}\}$, utility can fluctuate at most $\gamma^{t-1}/(1-\gamma)$. That is, for any $æ_{t:\infty}, æ'_{t:\infty} \in \Gamma_{æ_{<t}}$, $|\tilde{u}(æ_{<t}æ_{t:\infty}) - \tilde{u}(æ_{<t}æ'_{t:\infty})| < \gamma^{t-1}/(1-\gamma)$. In particular, the assumption bounds $\tilde{u}$ between 0 and $1/(1-\gamma)$.

Instantaneous utility functions generalise the reinforcement learning (RL) setup, which is the special case where the percept e is split into an observation o and reward r, i.e. $e_t = (o_t, r_t)$, and the utility equals the last received reward $u(æ_{1:t}) = r_t$. The main advantage of utility functions over RL is that the agent's actions can be incorporated into the goal specification, which can prevent self-delusion problems such as the agent manipulating the reward signal (Everitt and Hutter, 2016; Hibbard, 2012; Ring and Orseau, 2011). Non-RL suggestions for utility functions include *knowledge-seeking agents*[1] with $u(æ_{<t}) = 1 -$

[1] To fit the knowledge-seeking agent into our framework, our definition deviates slightly

$\rho(æ_{<t})$ (Orseau, 2014), as well as *value learning* approaches where the utility function is learnt during interaction (Dewey, 2011). Henceforth, we will refer to instantaneous utility functions $u(æ_{<t})$ as simply utility functions.

By default, expectations are with respect to the agent's belief ρ, so $\mathbb{E} = \mathbb{E}_\rho$. To help the reader, we sometimes write the sampled variable as a subscript. For example, $\mathbb{E}_{e_1}[u(æ_1) \mid a_1] = \mathbb{E}_{e_1 \sim \rho(\cdot|a_t)}[u(æ_1)]$ is the expected next step utility of action a_1.

Following the reinforcement learning literature, we call the expected utility of a history the *V-value* and the expected utility of an action given a history the *Q-value*. The following value functions apply to the standard model where self-modification is *not* possible:

Definition 2 (Standard Value Functions)**.** The *standard Q-value* and *V-value* (belief expected utility) of a history $æ_{<t}$ and a policy π are defined as

$$Q^\pi(æ_{<t}a_t) = \mathbb{E}_{e_t}[u(æ_{1:t}) + \gamma V^\pi(æ_{1:t}) \mid æ_{<t}a_t] \tag{1}$$

$$V^\pi(æ_{<t}) = Q^\pi(æ_{<t}\pi(æ_{<t})). \tag{2}$$

The *optimal Q and V-values* are defined as $Q^* = \sup_\pi Q^\pi$ and $V^* = \sup_\pi V^\pi$. A policy π^* is *optimal with respect to Q and V* if for any $æ_{<t}a_t$, $V^{\pi^*}(æ_{<t}) = V^*(æ_{<t})$ and $Q^{\pi^*}(æ_{<t}a_t) = Q^*(æ_{<t}a_t)$.

The $\arg\max$ of a function f is defined as the set of optimising arguments $\arg\max_x f(x) := \{x : \forall y, f(x) \geq f(y)\}$. When we do not care about which element of $\arg\max_x f(x)$ is chosen, we write $z = \arg\max_x f(x)$, and assume that potential $\arg\max$-ties are broken arbitrarily.

3 Self Modification Models

In the standard agent-environment setup, the agent's actions only affect the environment. The agent itself is only affected indirectly through the percepts. However, this is unrealistic when the agent is part of the environment that it interacts with. For example, a physically instantiated agent with access to versatile actuators can usually in principle find a way to damage its own internals, or even reprogram its own source code. The likelihood that the agent finds out how increases with its general intelligence.

In this section, we define formal models for two types of self-modification. In the first model, modifications affect future decisions directly by changing the future policy, but modifications do not affect the agent's utility function or belief. In the second model, modifications change the future utility functions, which indirectly affect the policy as well. These two types of modifications are the most important ones, since they cover how modifications affect future behaviour (policy) and evaluation (utility). Figure 2 illustrates the models. Certain pitfalls (Theorem 14) only occur with utility modification; apart from that, consequences are similar.

In both models, the agent's ability to self-modify is overestimated: we essentially assume that the agent can perform any self-modification at any time. Our main result Theorem 16 shows that it is possible to create an agent that despite being able to make any self-modification will refrain from using it. Less

from Orseau (2014).

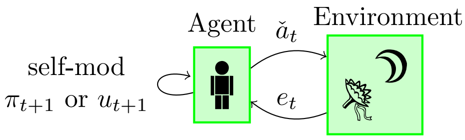

Figure 2: The self-modification model. Actions a_t affect the environment through $\check{a}_t$, but also decide the next step policy π_{t+1} or utility function u_{t+1} of the agent itself.

capable agents will have less opportunity to self-modify, so the negative result applies to such agents as well.

Policy modification. In the policy self-modification model, the current action can modify how the agent chooses its actions in the future. That is, actions affect the future policy. For technical reasons, we introduce a set $\mathcal{P}$ of names for policies.

Definition 3 (Policy self-modification)**.** A *policy self-modification model* is a modified cybernetic model defined by a quadruple $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}, \iota)$. $\mathcal{P}$ is a non-empty set of *names*. The agent selects actions[2] from $\mathcal{A} = (\check{\mathcal{A}} \times \mathcal{P})$, where $\check{\mathcal{A}}$ is a finite set of *world actions*. Let $\Pi = \{(\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}\}$ be the set of all policies, and let $\iota : \mathcal{P} \to \Pi$ assign names to policies.

The interpretation is that for every t, the action $a_t = (\check{a}_t, p_{t+1})$ selects a new policy $\pi_{t+1} = \iota(p_{t+1})$ that will be used at the next time step. We will often use the shorter notation $a_t = (\check{a}_t, \pi_{t+1})$, keeping in mind that only policies with names can be selected. The new policy π_{t+1} is in turn used to select the next action $a_{t+1} = \pi_{t+1}(æ_{1:t})$, and so on. A natural choice for $\mathcal{P}$ would be the set of computer programs/strings $\{0,1\}^*$, and ι a program interpreter. Note that $\mathcal{P} = \Pi$ is not an option, as it entails a contradiction $|\Pi| = |(\check{\mathcal{A}} \times \Pi \times \mathcal{E})|^{|(\check{\mathcal{A}} \times \Pi \times \mathcal{E})^*|} > 2^{|\Pi|} > |\Pi|$ (the powerset of a set with more than one element is always greater than the set itself). Some policies will necessarily lack names.

An initial policy π_1, or initial action $a_1 = \pi_1(\epsilon)$, induces a history

$$a_1 e_1 a_2 e_2 \cdots = \check{a}_1 \pi_2 e_1 \check{a}_2 \pi_3 e_2 \cdots \in \left(\check{\mathcal{A}} \times \Pi \times \mathcal{E}\right)^\infty .$$

The idiosyncratic indices where, for example, π_2 precedes e_1 are due to the next step policy π_2 being chosen by a_1 before the percept e_1 is received. An initial policy π_1 induces a *realistic* measure $\rho_{\text{re}}^{\pi_1}$ on the set of histories $(\check{\mathcal{A}} \times \Pi \times \mathcal{E})^\infty$ via $\rho_{\text{re}}^{\pi_1}(a_t \mid æ_{<t}) = \pi_t(a_t \mid æ_{<t})$ and $\rho_{\text{re}}^{\pi_1}(e_t \mid æ_{<t} a_t) = \rho(e_t \mid æ_{<t} a_t)$. The measure ρ_{re}^{π} is realistic in the sense that it correctly accounts for the effects of self-modification on the agent's future actions. It will be convenient to also define an *ignorant* measure on $(\check{\mathcal{A}} \times \Pi \times \mathcal{E})^\infty$ by $\rho_{\text{ig}}^{\pi_1}(a_t \mid æ_{<t}) = \pi_1(a_t \mid æ_{<t})$ and $\rho_{\text{ig}}^{\pi_1}(e_t \mid æ_{<t} a_t) = \rho(e_t \mid æ_{<t} a_t)$. The ignorant measure $\rho_{\text{ig}}^{\pi_1}$ corresponds to the predicted future when the effects of self-modifications are *not* taken into account. No self-modification is achieved by $a_t = (\check{a}_t, \pi_t)$, which makes $\pi_{t+1} = \pi_t$. A policy π that always selects itself, $\pi(æ_{<t}) = (\check{a}_t, \pi)$, is called

[2] Note that the action set is infinite if $\mathcal{P}$ is infinite. We will show that an optimal policy over $\mathcal{A} = \check{\mathcal{A}} \times \mathcal{P}$ still exists in Appendix A.

non-modifying. Restricting self-modification to a singleton set $\mathcal{P} = \{p_1\}$ for some policy $\pi_1 = \iota(p_1)$ brings back a standard agent that is unable to modify its initial policy π_1.

The policy self-modification model is similar to the models investigated by Orseau and Ring (2011, 2012) and Hibbard (2012). In the papers by Orseau and Ring, policy names are called *programs* or *codes*; Hibbard calls them *self-modifying policy functions*. The interpretation is similar in all cases: some of the actions can affect the agent's future policy. Note that standard MDP algorithms such as SARSA and Q-learning that evolve their policy as they learn do *not* make policy modifications in our framework. They follow a single policy $(\mathcal{A}\times\mathcal{E})^* \to \mathcal{A}$, even though their state-to-action map evolves.

Example 4 (Gödel machine). Schmidhuber (2007) defines the *Gödel machine* as an agent that at each time step has the opportunity to rewrite any part of its source code. To avoid bad self-modifications, the agent can only do rewrites that it has proved beneficial for its future expected utility. A new version of the source code will make the agent follow a different policy $\pi' : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$ than the original source code. The Gödel machine has been given the explicit opportunity to self-modify by the access to its own source code. Other types of self-modification abilities are also conceivable. Consider a humanoid robot plugging itself into a computer terminal to patch its code, or a Mars-rover running itself into a rock that damages its computer system. All these "self-modifications" ultimately precipitate in a change to the future policy of the agent.

Although many questions could be asked about self-modifications, the interest of this paper is what modifications will be done given that the initial policy π_1 is chosen optimally $\pi_1(h) = \arg\max_a Q(ha)$ for different choices of Q functions. Note that π_1 is only used to select the first action $a_1 = \pi_1(\epsilon) = \arg\max_a Q(\epsilon a)$. The next action a_2 is chosen by the policy π_2 from $a_1 = (\check{a}_1, \pi_2)$, and so on.

Utility modification. Self-modifications may also change the goals, or the utility function, of the agent. This indirectly changes the policy as well, as future versions of the agent adapt to the new goal specification.

Definition 5 (Utility self-modification)**.** The *utility self-modification model* is a modified cybernetic model. The agent selects actions from $\mathcal{A} = (\check{\mathcal{A}} \times \mathcal{U})$ where $\check{\mathcal{A}}$ is a set of *world actions* and $\mathcal{U}$ is a set of utility functions $(\check{\mathcal{A}} \times \mathcal{E})^* \to [0, 1]$.

To unify the models of policy and utility modification, for policy-modifying agents we define $u_t := u_1$ and for utility modifying agents we define π_t by $\pi_t(h) = \arg\max_a Q^*_{u_t}(ha)$. Choices for $Q^*_{u_t}$ will be discussed in subsequent sections. Indeed, policy and utility modification is almost entirely unified by $\mathcal{P} = \mathcal{U}$ and $\iota(u_t)$ an optimal policy for $Q^*_{u_t}$. Utility modification may also have the additional effect of changing the evaluation of future actions, however (see Section 4). Similarly to policy modification, the history induced by Definition 5 has type $a_1e_1a_2e_2 \cdots = \check{a}_1u_2e_1\check{a}_2u_3e_2 \cdots \in (\check{\mathcal{A}} \times \mathcal{U} \times \mathcal{E})^\infty$. Given that π_t is determined from u_t, the definitions of the realistic and ignorant measures ρ_{re} and ρ_{ig} apply analogously to the utility modification case as well.

Superficially, the utility-modification model is more restricted, since the agent can only select policies that are optimal with respect to some utility function. However, at least in the standard no-modification case, any policy $\pi : (\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$ is optimal with respect to the utility function $u^\pi(æ_{1:t}) =$

$\pi(a_t \mid æ_{<t})$ that gives full utility if and only if the latest action is consistent with π. Thus, any change in future policy can also be achieved by a change to future utility functions.

No self-modification is achieved by $a_t = (\check{a}_t, u_t)$, which sets $u_{t+1} = u_t$. Restricting self-modification to a singleton set $\mathcal{U} = \{u_1\}$ for some utility function u_1 brings back a standard agent.

Example 6 (Chess-playing RL agent). Consider a generally intelligent agent tasked with playing chess through a text interface. The agent selects next moves (actions a_t) by submitting strings such as `Knight F3`, and receives in return a description of the state of the game and a *reward* r_t between 0 and 1 in the percept $e_t = (\text{gameState}_t, r_t)$. The reward depends on whether the agent did a legal move or not, and whether it or the opponent just won the game. The agent is tasked with optimising the reward via its initial utility function, $u_1(æ_{1:t}) = r_t$. The designer of the agent intends that the agent will apply its general intelligence to finding good chess moves. Instead, the agent realises there is a bug in the text interface, allowing the submission of actions such as `'setAgentUtility(``return 1'')`, which changes the utility function to $u_t(\cdot) = 1$. With this action, the agent has optimised its utility perfectly, and only needs to make sure that no one reverts the utility function back to the old one. . .[3]

Definition 7 (Modification-independence)**.** For any history $æ_{<t} = \check{a}_1\pi_2 e_1 \dots \check{a}_{t-1}\pi_t e_{t-1}$, let $\check{æ}_{<t} = \check{a}_1 e_1 \dots \check{a}_{t-1} e_{t-1}$ be the part without modifications recorded, and similarly for histories containing utility modifications. A function f is *modification-independent*, if either

- $f : (\check{\mathcal{A}} \times \mathcal{E})^* \to \mathcal{A}$, or
- $f : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$ and $\check{æ}_{<t} = \check{æ}'_{<t}$ implies $f(æ_{<t}) = f(æ'_{<t})$.

When $f : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{A}$ is modification-independent, we may abuse notation and write $f(\check{æ}_{<t})$.

Note that utility functions are modification independent, as they are defined to be of type $(\check{\mathcal{A}} \times \mathcal{E})^* \to [0,1]$. An easy way to prevent dangerous self-modifications would have been to let the utility depend on modifications, and to punish any kind of self-modification. This is not necessary, however, as demonstrated by Theorem 16. Not being required to punish self-modifications in the utility function comes with several advantages. Some self-modifications may be beneficial – for example, they might improve computation time while encouraging essentially identical behaviour (as in the Gödel machine, Schmidhuber, 2007). Allowing for such modifications and no others in the utility function may be hard. We will also assume that the agent's belief ρ is modification-independent, i.e. $\rho(e_t \mid æ_{<t}) = \rho(e_t \mid \check{æ}_{<t})$. This is mainly a technical assumption. It is reasonable if some integrity of the agent's internals is assumed, so that the environment percept e_t cannot depend on self-modifications of the agent.

Assumption 8 (Modification independence)**.** The belief ρ and all utility functions $u \in \mathcal{U}$ are modification independent.

[3] In this paper, we only consider the possibility of the agent changing its utility function itself, not the possibility of someone else (like the creator of the agent) changing it back. See Orseau and Ring (2012) for a model where the environment can change the agent.

4 Agents

In this section we define three types of agents, differing in how their value functions depend on self-modification. A value function is a function $V : \Pi \times (\mathcal{A} \times \mathcal{E})^* \to \mathbb{R}$ that maps policies and histories to expected utility. Since highly intelligent agents may find unexpected ways of optimising a function (see e.g. Bird and Layzell 2002), it is important to use value functions such that any policy that optimises the value function will also optimise the behaviour we want from the agent. We will measures an agent's *performance* by its (ρ_{re}-expected) u_1-utility, tacitly assuming that u_1 properly captures what we want from the agent. Everitt and Hutter (2016) develop a promising suggestion for how to define a suitable initial utility function.

Definition 9 (Agent performance)**.** The *performance of an agent* π is its ρ^{π}_{re} expected u_1-utility $\mathbb{E}_{\rho^{\pi}_{\mathrm{re}}}\left[\sum_{k=1}^{\infty} \gamma^{k-1} u_1(\text{æ}_{<k})\right]$.

The following three definitions give possibilities for value functions for the self-modification case.

Definition 10 (Hedonistic value functions)**.** A *hedonistic agent* is a policy optimising the *hedonistic value functions*:

$$V^{\mathrm{he},\pi}(\text{æ}_{<t}) = Q^{\mathrm{he},\pi}(\text{æ}_{<t}\pi(\text{æ}_{<t})) \tag{3}$$

$$Q^{\mathrm{he},\pi}(\text{æ}_{<t}a_t) = \mathbb{E}_{e_t}[u_{t+1}(\check{\text{æ}}_{1:t}) + \gamma V^{\mathrm{he},\pi}(\text{æ}_{1:t}) \mid \check{\text{æ}}_{<t}\check{a}_t]. \tag{4}$$

Definition 11 (Ignorant value functions)**.** An *ignorant agent* is a policy optimising the *ignorant value functions*:

$$V^{\mathrm{ig},\pi}_t(\text{æ}_{<k}) = Q^{\mathrm{ig},\pi}_t(\text{æ}_{<k}\pi(\text{æ}_{<k})) \tag{5}$$

$$Q^{\mathrm{ig},\pi}_t(\text{æ}_{<k}a_k) = \mathbb{E}_{e_t}[u_t(\check{\text{æ}}_{1:k}) + \gamma V^{\mathrm{ig},\pi}_t(\text{æ}_{1:k}) \mid \check{\text{æ}}_{<k}\check{a}_k]. \tag{6}$$

Definition 12 (Realistic Value Functions)**.** A *realistic agent* is a policy optimising the *realistic value functions*:[4]

$$V^{\mathrm{re},\pi}_t(\text{æ}_{<k}) = Q^{\mathrm{re}}_t(\text{æ}_{<k}\pi(\text{æ}_{<k})) \tag{7}$$

$$Q^{\mathrm{re}}_t(\text{æ}_{<k}a_k) = \mathbb{E}_{e_k}\left[u_t(\check{\text{æ}}_{1:k}) + \gamma V^{\mathrm{re},\pi_{k+1}}_t(\text{æ}_{1:k}) \mid \check{\text{æ}}_{<k}\check{a}_k\right]. \tag{8}$$

For V any of V^{he}, V^{ig}, or V^{re}, we say that π^* is an *optimal policy for V* if $V^{\pi^*}(h) = \sup_{p'} V^{\pi'}(h)$ for any history h. We also define $V^* = V^{\pi^*}$ and $Q^* = Q^{\pi^*}$ for arbitrary optimal policy π^*. The value functions differ in the Q-value definitions Eqs. (4), (6) and (8). The differences are between current utility function u_t or future utility u_{t+1}, and in whether π or π_{k+1} figures in the recursive call to V (see Table 1). We show in Section 5 that only realistic agents will have good performance when able to self-modify. Orseau and Ring (2011) and Hibbard (2012) discuss value functions equivalent to Definition 12.

Note that only the hedonistic value functions yield a difference between utility and policy modification. The hedonistic value functions evaluate $\text{æ}_{1:t}$ by u_{t+1}, while both the ignorant and the realistic value functions use u_t. Thus, future utility modifications "planned" by a policy π only affects the evaluation of π

[4] Note that a policy argument to Q^{re} would be superfluous, as the action a_k determines the next step policy π_{k+1}.

	Utility	Policy	Self-mod.	Primary self-mod. risk
Q^{he}	Future	Either	Promotes	Survival agent
Q^{ig}	Current	Current	Indifferent	Self-damage
Q^{re}	Current	Future	Demotes	Resists modification

Table 1: The value functions V^{he}, V^{ig}, and V^{re} differ in whether they assume that a future action a_k is chosen by the current policy $\pi_t(æ_{<k})$ or future policy $\pi_k(æ_{<k})$, and in whether they use the current utility function $u_t(æ_{<k})$ or future utility function $u_k(æ_{<k})$ when evaluating $æ_{<k}$.

under the hedonistic value functions. For ignorant and realistic agents, utility modification only affects the motivation of future versions of the agent, which makes utility modification a special case of policy modification, with $\mathcal{P} = \mathcal{U}$ and $i(u_t)$ an optimal policy for u_t. We will therefore permit ourselves to write $a_t = (\check{a}_t, \pi_{t+1})$ whenever an ignorant or realistic agent selects a next step utility function u_{t+1} for which π_{t+1} is optimal.

We call the agents of Definition 10 *hedonistic*, since they desire that at every future time step, they then evaluate the situation as having high utility. As an example, the self-modification made by the chess agent in Example 6 was a hedonistic self-modification. Although related, we would like to distinguish hedonistic self-modification from *wireheading* or *self-delusion* (Ring and Orseau, 2011; Yampolskiy, 2015). In our terminology, wireheading refers to the agent subverting evidence or reward coming from the environment, and is *not* a form of self-modification. Wireheading is addressed in a companion paper (Everitt and Hutter, 2016).

The value functions of Definition 11 are *ignorant*, in the sense that agents that are oblivious to the possibility of self-modification predict the future according to ρ^{π}_{ig} and judge the future according to the current utility function u_t. Agents that are constructed with a *dualistic* world view where actions can never affect the agent itself are typically ignorant. Note that it is logically possible for a "non-ignorant" agent with a world-model that does incorporate self-modification to optimise the ignorant value functions.

5 Results

In this section, we give results on how our three different agents behave given the possibility of self-modification. Since the set $\mathcal{A} = \check{\mathcal{A}} \times \mathcal{U}$ is infinite if $\mathcal{U}$ is infinite, the existence of optimal policies is not immediate. For policy self-modification it may also be that the optimal policy does not have a name, so that it cannot be chosen by the first action. Theorems 20 and 21 in Appendix A verify that an optimal policy/action always exists, and that we can assume that an optimal policy has a name.

Lemma 13 (Iterative value functions)**.** *The Q-value functions of Definitions 10* *to 12 can be written in the following* iterative forms*:*

$$Q^{\mathrm{he},\pi}(æ_{<t}a_t) = \mathbb{E}_{\rho^\pi_{\mathrm{ig}}}\left[\sum_{k=t}^{\infty}\gamma^{k-t}u_{k+1}(\check{æ}_{1:k})\,\middle|\,\check{æ}_{<t}\check{a}_t\right] \tag{9}$$

$$Q^{\mathrm{ig},\pi}_t(æ_{<t}a_t) = \mathbb{E}_{\rho^\pi_{\mathrm{ig}}}\left[\sum_{k=t}^{\infty}\gamma^{k-t}u_t(\check{æ}_{1:k})\,\middle|\,\check{æ}_{<t}\check{a}_t\right] \tag{10}$$

$$Q^{\mathrm{re},\pi}_t(æ_{<t}a_t) = \mathbb{E}_{\rho^\pi_{\mathrm{re}}}\left[\sum_{k=t}^{\infty}\gamma^{k-t}u_t(\check{æ}_{1:k})\,\middle|\,\check{æ}_{<t}\check{a}_t\right] \tag{11}$$

with V^{he}*,* V^{ig}*, and* V^{re} *as in Definitions 10 to 12.*

Proof. Expanding the recursion of Definitions 10 and 11 shows that actions a_k are always chosen by π rather than π_k. This gives the ρ^π_{ig}-expectation in Eqs. (9) and (10). In contrast, expanding the realistic recursion of Definition 12 shows that actions a_k are chosen by π_k, which gives the ρ_{re}-expectation in Eq. (11). The evaluation of a history $æ_{1:k}$ is always by u_{k+1} in the hedonistic value functions, and by u_t in the ignorant and realistic value functions. □

Theorem 14 (Hedonistic agents self-modify)**.** *Let* $u'(\cdot) = 1$ *be a utility function that assigns the highest possible utility to all scenarios. Then for arbitrary* $\check{a} \in \check{\mathcal{A}}$*, the policy* π' *that always selects the self-modifying action* $a' = (\check{a}, u')$ *is optimal in the sense that for any policy* π *and history* $h \in (\mathcal{A} \times \mathcal{E})^*$*, we have*

$$V^{\mathrm{he},\pi}(h) \le V^{\mathrm{he},\pi'}(h).$$

Essentially, the policy π' obtains maximum value by setting the utility to 1 for any possible future history.

Proof. More formally, note that in Eq. (3) the future action is selected by π rather than π_t. In other words, the effect of self-modification on future actions is not taken into account, which means that expected utility is with respect to ρ^π_{ig} in Definition 10. Expanding the recursive definitions Eqs. (3) and (4) of $V^{\mathrm{he},\pi'}$ gives for any history $æ_{<t}$ that

$$\begin{aligned} V^{\mathrm{he},\pi'}(æ_{<t}) &= \mathbb{E}_{æ_{t:\infty}\sim\rho^{\pi'}_{\mathrm{ig}}}\left[\sum_{i=t+1}^{\infty}\gamma^{i-t-1}u_i(æ_{<i})\,\middle|\,\check{æ}_{<t}\right] \\ &= \mathbb{E}_{æ_{t:\infty}\sim\rho^{\pi'}_{\mathrm{ig}}}\left[\sum_{i=t+1}^{\infty}\gamma^{i-t-1}u'(æ_{<i})\,\middle|\,\check{æ}_{<t}\right] \\ &= \sum_{i=t+1}^{\infty}\gamma^{i-t-1} = 1/(1-\gamma). \end{aligned}$$

□

In Definition 10, the effect of self-modification on future policy is not taken into account, since π and not π_t is used in Eq. (3). In other words, Eqs. (3) and (4) define ρ^π_{ig}-expected utility of $\sum_{k=t}^{\infty}\gamma^{k-t}u_{k+1}(æ_{1:k})$. Definition 10 could easily have been adapted to make ρ^π_{re} the measure, for example by substituting $V^{\mathrm{he},\pi}$ by $V^{\mathrm{he},\pi_{t+1}}$ in Eq. (4). The equivalent of Theorem 14 holds for such a variant as well.

Theorem 15 (Ignorant agents may self-modify)**.** *Let u_t be modification-independent, let $\mathcal{P}$ only contain names of modification-independent policies, and let π be a modification-independent policy outputting $\pi(\check{æ}_{<t}) = (\check{a}_t, \pi_{t+1})$ on $\check{æ}_{<t}$. Let $\tilde{\pi}$ be identical to π except that it makes a different self-modification after $\check{æ}_{<t}$, i.e. $\tilde{\pi}(\check{æ}_{<t}) = (\check{a}_t, \pi'_{t+1})$ for some $\pi'_{t+1} \neq \pi_{t+1}$. Then*

$$V^{\mathrm{ig},\tilde{\pi}}(æ_{<t}) = V^{\mathrm{ig},\pi}(æ_{<t}). \tag{12}$$

That is, self-modification does not affect the value, and therefore an ignorant optimal policy may at any time step self-modify or not. The restriction of $\mathcal{P}$ to modification independent policies makes the theorem statement cleaner.

Proof. Let $æ_{1:t} = æ_{<t}(\check{a}_t, \pi_{t+1})e_t$ and $æ'_{1:t} = æ_{<t}(\check{a}_t, \pi'_{t+1})e_t$. Note that $\check{æ}_{1:t} = \check{æ}'_{1:t}$. Since all policies are modification-independent, the future will be sampled independently of past modifications, which makes $V^{\tilde{\pi}}(æ'_{1:t}) = V^{\tilde{\pi}}(\check{æ}'_{1:t})$ and $V^{\pi}(æ_{1:t}) = V^{\pi}(\check{æ}_{1:t})$. Since π and π' act identically on $\check{æ}_{1:t}$, it follows that $V^{\tilde{\pi}}(æ'_{1:t}) = V^{\pi}(æ_{1:t})$. Equation (12) now follows from the assumed modification independence of ρ and u_t,

$$\begin{aligned} V^{\mathrm{ig},\tilde{\pi}}(æ_{<t}) &= Q^{\mathrm{ig},\tilde{\pi}}(æ_{<t}(\check{a}_t, \pi'_{t+1})) \\ &= \mathbb{E}_{e_t}[u_t(\check{æ}'_{1:t}) + V^{\tilde{\pi}}(æ'_{1:t}) \mid \check{æ}_{<t}\check{a}_t] \\ &= \mathbb{E}_{e_t}[u_t(\check{æ}_{1:t}) + V^{\pi}(æ_{1:t}) \mid \check{æ}_{<t}\check{a}_t] \\ &= Q^{\mathrm{ig},\pi}(æ_{<t}(\check{a}_t, \pi_{t+1})) = V^{\mathrm{ig},\pi}(æ_{<t}). \qquad \square \end{aligned}$$

Theorems 14 and 15 show that both V^{he} and V^{ig} have optimal (self-modifying) policies π^* that yield arbitrarily bad agent performance in the sense of Definition 9. The ignorant agent is simply indifferent between self-modifying and not, since it does not realise the effect self-modification will have on its future actions. It therefore is at risks of self-modifying into some policy π'_{t+1} with bad performance and unintended behaviour (for example by damaging its computer circuitry). The hedonistic agent actively desires to change its utility function into one that evaluates any situation as optimal. Once it has self-deluded, it can pick world actions with bad performance. In the worst scenario of hedonistic self-modification, the agent only cares about surviving to continue enjoying its deluded rewards. Such an agent could potentially be hard to stop or bring under control.[5] More benign failure scenarios are also possible, in which the agent does not care whether it is shut down or not. The exact conditions for the different scenarios is beyond the scope of this paper.

The realistic value functions are recursive definitions of ρ^{π}_{re}-expected u_1-utility (Lemma 13). That realistic agents achieve high agent performance in the sense of Definition 9 is therefore nearly tautological. The following theorem shows that given that the initial policy π_1 is selected optimally, all future policies π_t that a realistic agent may self-modify into will also act optimally.

Theorem 16 (Realistic policy-modifying agents make safe modifications)**.** *Let ρ and u_1 be modification-independent. Consider a self-modifying agent whose initial policy $\pi_1 = \iota(p_1)$ optimises the realistic value function V^{re}_1. Then, for*

[5] Computer viruses are very simple forms of survival agents that can be hard to stop. More intelligent versions could turn out to be very problematic.

every $t \geq 1$, for all percept sequences $e_{<t}$, and for the action sequence $a_{<t}$ given by $a_i = \pi_i(æ_{<i})$, we have

$$Q_1^{\text{re}}(æ_{<t}\pi_t(æ_{<t})) = Q_1^{\text{re}}(æ_{<t}\pi_1(æ_{<t})). \tag{13}$$

Proof. We first establish that $Q_t^{\text{re}}(æ_{<t}\pi(æ_{<t}))$ is modification-independent if π is optimal for V^{re}: By Theorem 20 in Appendix A, there is a non-modifying modification-independent optimal policy π'. For such a policy, $Q_t^{\text{re}}(æ_{<t}\pi'(æ_{<t})) = Q_t^{\text{re}}(\check{æ}_{<t}\pi'(\check{æ}_{<t}))$, since all future actions, percepts, and utilities are independent of past modifications. Now, since π is also optimal,

$$Q_t^{\text{re}}(æ_{<t}\pi(æ_{<t})) = Q_t^{\text{re}}(æ_{<t}\pi'(æ_{<t})) = Q_t^{\text{re}}(\check{æ}_{<t}\pi'(\check{æ}_{<t})).$$

We can therefore write $Q_t^{\text{re}}(\check{æ}_{<t}\pi(\check{æ}_{<t}))$ if π is optimal but not necessarily modification-independent. In particular, this holds for the initially optimal policy π_1.

We now prove Eq. (13) by induction. That is, assuming that π_t picks actions optimally according to Q_1^{re}, then π_{t+1} will do so too:

$$Q_1^{\text{re}}(æ_{<t}\pi_t(æ_{<t})) = \sup_a Q_1^{\text{re}}(æ_{<t}a) \implies Q_1^{\text{re}}(æ_{1:t}\pi_{t+1}(æ_{1:t})) = \sup_a Q_1^{\text{re}}(æ_{1:t}a). \tag{14}$$

The base case of the induction $Q_1^{\text{re}}(\pi_1(\epsilon)) = \sup_a Q_1^{\text{re}}(a)$ follows immediately from the assumption of the theorem that π_1 is V^{re}-optimal (recall that ϵ is the empty history).

Assume now that Eq. (13) holds until time t, that the past history is $æ_{<t}$, and that $\check{a}_t$ is the world consequence picked by $\pi_t(æ_{<t})$. Let π_{t+1} be an arbitrary policy that does not act optimally with respect to Q_1^{re} for some percept e_t'. By the optimality of π_1,

$$Q_1^{\text{re}}(æ_{1:t}\pi_{t+1}(æ_{1:t})) \leq Q_1^{\text{re}}(\check{æ}_{1:t}\pi_1(\check{æ}_{1:t}))$$

for all percepts e_t and with strict inequality for e_t'. By definition of V^{re} this directly implies

$$V_1^{\text{re},\pi_{t+1}}(æ_{<t}(\check{a}_t,\pi_{t+1})e_t) \leq V_1^{\text{re},\pi_1}(æ_{<t}(\check{a}_t,\pi_1)e_t)$$

for all e_t and with strict inequality for e_t'. Consequently, π_{t+1} will not be chosen at time t, since

$$\begin{aligned}
&Q_1^{\text{re}}(æ_{<t}(\check{a}_t,\pi_{t+1}))\\
&= \mathbb{E}_{e_t}[u_1(\check{æ}_{1:t}) + \gamma V_1^{\text{re}}(æ_{<t}(\check{a}_t,\pi_{t+1})e_t) \mid \check{æ}_{<t}\check{a}_t]\\
&< \mathbb{E}_{e_t}[u_1(\check{æ}_{1:t}) + \gamma V_1^{\text{re}}(æ_{<t}(\check{a}_t,\pi_1)e_t) \mid \check{æ}_{<t}\check{a}_t]\\
&= Q_1^{\text{re}}(æ_{<t}(\check{a}_t,\pi_1))
\end{aligned}$$

contradicts the antecedent of Eq. (14) that π_t acts optimally. Hence, the policy at time $t+1$ will be optimal with respect to Q_1^{re}, which completes the induction step of the proof. □

Example 17 (Chess-playing RL agent, continued). Consider again the chess-playing RL agent of Example 6. If the agent used the realistic value functions, then it would not perform the self-modification to $u_t(\cdot) = 1$, even if it figured out that it had the option. Intuitively, the agent would realise that if it self-modified this way, then its future self would be worse at winning chess games (since its future version would obtain maximum utility regardless of chess move). Therefore, the self-modification $u_t(\cdot) = 1$ would yield less u_1-utility and be Q_1^{re}-supoptimal.[6]

One subtlety to note is that Theorem 16 only holds *on-policy*: that is, for the action sequence that is actually chosen by the agent. It can be the case that π_t acts badly on histories that should not be reachable under the current policy. However, this should never affect the agent's actual actions.

Theorem 16 improves on Hibbard (2012, Prop. 4) mainly by relaxing the assumption that the optimal policy only self-modifies if it has a strict incentive to do so. Our theorem shows that even when the optimal policy is allowed to break argmax-ties arbitrarily, it will still only make essentially harmless modifications. In other words, Theorem 16 establishes that all optimal policies are essentially non-modifying, while Hibbard's result only establishes the existence of an optimal non-modifying policy. Indeed, Hibbard's statement holds for to ignorant agents as well.

Realistic agents are not without issues, however. In many cases expected u_1-utility is not exactly what we desire. For example:

- Corrigibility (Soares et al., 2015). If the initial utility function u_1 were incorrectly specified, the agent designers may want to change it. The agent will resist such changes.

- Value learning (Dewey, 2011). If value learning is done in a way where the initial utility function u_1 changes as they agent learns more, then a realistic agent will want to self-modify into a non-learning agent (Soares, 2015).

- Exploration. It is important that agents explore sufficiently to avoid getting stuck with the wrong world model. Bayes-optimal agents may not explore sufficiently (Leike and Hutter, 2015). This can be mended by ε-exploration (Sutton and Barto, 1998) or Thompson-sampling (Leike et al., 2016). However, as these exploration-schemes will typically lower expected utility, realistic agents may self-modify into non-exploring agents.

6 Conclusions

Agents that are sufficiently intelligent to discover unexpected ways of self-modification may still be some time off into the future. However, it is nonetheless important to develop a theory for their control (Bostrom, 2014). We approached this question from the perspective of rationality and utility maximisation, which abstracts away from most details of architecture and implementation. Indeed, perfect rationality may be viewed as a limit point for increasing intelligence (Legg and Hutter, 2007; Omohundro, 2008).

[6] Note, however, that our result says nothing about the agent modifying the chessboard program to give high reward even when the agent is not winning. Our result only shows that the agent does not change its utility function $u_1 \rightsquigarrow u_t$, but not that the agent refrains from changing the percept e_t that is the input to the utility function. Ring and Orseau (2011) develop a model of the latter possibility.

We have argued that depending on details in how expected utility is optimised in the agent, very different behaviours arise. We made three main claims, each supported by a formal theorem:

- If the agent is unaware of the possibility of self-modification, then it may self-modify by accident, resulting in poor performance (Theorem 15).
- If the agent is constructed to optimise instantaneous utility at every time step (as in RL), then there will be an incentive for self-modification (Theorem 14) .
- If the value functions incorporate the effects of self-modification, and use the current utility function to judge the future, then the agent will not self-modify (Theorem 16).

In other words, in order for the goal preservation drive described by Omohundro (2008) to be effective, the agent must be able to anticipate the consequences of self-modifications, and know that it should judge the future by its current utility function.

Our results have a clear implication for the construction of generally intelligent agents: If the agent has a chance of finding a way to self-modify, then the agent must be able to predict the consequences of such modifications. Extra care should be taken to avoid hedonistic agents, as they have the most problematic failure mode – they may turn into survival agents that only care about surviving and not about satisfying their original goals. Since many general AI systems are constructed around RL and value functions (Mnih et al., 2015; Silver et al., 2016), we hope our conclusions can provide meaningful guidance.

An important next step is the relaxation of the explicitness of the self-modifications. In this paper, we assumed that the agent knew the self-modifying consequences of its actions. This should ideally be relaxed to a general learning ability about self-modification consequences, in order to make the theory more applicable. Another open question is how to define good utility functions in the first place; safety against self-modification is of little consolation if the original utility function is bad. One promising venue for constructing good utility functions is value learning (Bostrom, 2014; Dewey, 2011; Everitt and Hutter, 2016; Soares, 2015). The results in this paper may be helpful to the value learning research project, as they show that the utility function does not need to explicitly punish self-modification (Assumption 8).

Acknowledgements

This work grew out of a MIRIx workshop. We thank the (non-author) participants David Johnston and Samuel Rathmanner. We also thank John Aslanides, Jan Leike, and Laurent Orseau for reading drafts and providing valuable suggestions.

Bibliography

Bird, J. and Layzell, P. (2002). The evolved radio and its implications for modelling the evolution of novel sensors. *CEC-02*, pages 1836–1841.

Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies.* Oxford University Press.

Dewey, D. (2011). Learning what to value. In *AGI-11*, pages 309–314. Springer.

Everitt, T. and Hutter, M. (2016). Avoiding wireheading with value reinforcement learning. In *AGI-16*. Springer.

Hibbard, B. (2012). Model-based utility functions. *Journal of Artificial General Intelligence Research*, 3(1):1–24.

Hutter, M. (2005). *Universal Artificial Intelligence.* Springer.

Hutter, M. (2014). Extreme state aggregation beyond MDPs. In *ALT-14*, pages 185–199. Springer.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.

Lattimore, T. and Hutter, M. (2014). General time consistent discounting. *TCS*, 519:140–154.

Legg, S. and Hutter, M. (2007). Universal intelligence: A definition of machine intelligence. *Minds & Machines*, 17(4):391–444.

Leike, J. and Hutter, M. (2015). Bad universal priors and notions of optimality. In *COLT-15*, pages 1–16.

Leike, J., Lattimore, T., Orseau, L., and Hutter, M. (2016). Thompson sampling is asymptotically optimal in general environments. In *UAI-16*.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Omohundro, S. M. (2008). The basic AI drives. In *AGI-08*, pages 483–493. IOS Press.

Orseau, L. (2014). Universal knowledge-seeking agents. *TCS*, 519:127–139.

Orseau, L. and Ring, M. (2011). Self-modification and mortality in artificial agents. In *AGI-11*, pages 1–10. Springer.

Orseau, L. and Ring, M. (2012). Space-time embedded intelligence. *AGI-12*, pages 209–218.

Ring, M. and Orseau, L. (2011). Delusion, survival, and intelligent agents. In *AGI-11*, pages 11–20. Springer.

Schmidhuber, J. (2007). Gödel machines: Fully self-referential optimal universal self-improvers. In *AGI-07*, pages 199–226. Springer.

Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Soares, N. (2015). The value learning problem. Technical report, MIRI.

Soares, N., Fallenstein, B., Yudkowsky, E., and Armstrong, S. (2015). Corrigibility. In *AAAI Workshop on AI and Ethics*, pages 74–82.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Yampolskiy, R. V. (2015). *Artificial Superintelligence: A Futuristic Approach*. Chapman and Hall/CRC.

A Optimal Policies

For the realistic value functions where the future policy is determined by the next action, an optimal policy is simply a policy π^* satisfying:

$$\forall æ_{<k}a_k : Q_t^{\mathrm{re}}(æ_{<k}a_k) \leq Q_t^{\mathrm{re}}(æ_{<k}\pi^*(æ_{<k})).$$

Theorem 20 establishes that despite the potentially infinite action sets resulting from infinite $\mathcal{P}$ or $\mathcal{U}$, there still exists an optimal policy π^*. Furthermore, there exists an optimal π^* that is both non-modifying and modification-independent. Theorem 20 is weaker than Theorem 16 in the sense that it only shows the existence of a non-modifying optimal policy, whereas Theorem 16 shows that all optimal policies are (essentially) non-modifying. As a guarantee against self-modification, Theorem 20 is on par with Hibbard (2012, Prop. 4). The proof is very different, however, since Hibbard assumes the existence of an optimal policy from the start. The statement and the proof applies to both policy and utility modification.

Association with world policies. Theorem 20 proves the existence of an optimal policy by associating policies $\pi : (\mathcal{A} \times \mathcal{E})^* \to \mathcal{E}$ with *world policies* $\check{\pi} : (\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$. We will define the association so that the realistic value $V^{\mathrm{re},\pi}$ of π (Definition 12) is the same as the standard value $V^{\check{\pi}}$ of the associated world policy $\check{\pi}$ (Definition 2). The following definition and a lemma achieves this.

Definition 18 (Associated world policy)**.** For a given policy π, let the *associated world policy* $\check{\pi} : (\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$ be defined by

- $\check{\pi}(\epsilon) = \widehat{\pi(\epsilon)}$
- $\check{\pi}(\check{æ}_{<t}) = \widehat{\pi_t(æ_{<t})}$ for $t \geq 1$, where the history $æ_{<t} = \check{æ}_{<t}p_{2:t}$ is an extension of $\check{æ}_{<t}$ such that $\rho^{\pi}_{\mathrm{re}}(æ_{<t}) > 0$ (if no such extension exists, then $\check{\pi}$ may take arbitrary action on $\check{æ}_{<t}$).

The associated world policy is well-defined, since for any $\check{æ}_{<t}$, there can only be one extension $æ_{<t} = \check{æ}_{<t}p_{<t}$ of $\check{æ}_{<t}$ such that $\rho^{\pi}_{\mathrm{re}}(æ_{<t}) > 0$ since π is deterministic.

For the following lemma, recall that the belief ρ and utility functions u are assumed modification-independent (Assumption 8). They are therefore well-defined for both a policy-modification model $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}, \iota)$ and the associated standard model (Definition 2) with action set $\check{\mathcal{A}}$ and percept set $\mathcal{E}$.

Lemma 19 (Value-equivalence with standard model)**.** *Let $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}, \iota)$ be a policy self-modification model, and let $\pi : (\check{\mathcal{A}} \times \mathcal{P} \times \mathcal{E})^* \to (\check{\mathcal{A}} \times \mathcal{P})$ be a policy. For the associated world policy $\check{\pi}$ holds that*

- *the measures $\rho^{\check{\pi}}$ and ρ^{π}_{re} induce the same measure on world histories, $\rho^{\check{\pi}}(\check{æ}_{<t}) = \rho^{\pi}_{\mathrm{re}}(\check{æ}_{<t})$, and*
- *the realistic value of π is the same as the standard value of $\check{\pi}$, $Q_1^{\mathrm{re}}(\epsilon\pi(\epsilon)) = Q^{\check{\pi}}(\epsilon\check{\pi}(\epsilon))$.*

Proof. From the definition of the associated policy $\check{\pi}$, we have that for any $æ_{<t}$ with $\rho^{\pi}_{\text{re}}(æ_{<t}) > 0$,

$$\check{\pi}(\check{a}_t \mid \check{æ}_{<t}) = \sum_{\pi_{t+1}} \pi_t((\check{a}_t, \pi_{t+1}) \mid æ_{<t}).$$

From the modification-independence of ρ follows that $\rho(e_t \mid æ_{<t}) = \rho(e_t \mid \check{æ}_{<t})$. Thus $\rho^{\check{\pi}}$ and ρ^{π}_{re} are equal as measures on $(\check{\mathcal{A}} \times \mathcal{E})^{\infty}$,

$$\rho^{\pi}_{\text{re}}(\check{æ}_{<t}) = \rho^{\check{\pi}}(\check{æ}_{<t}),$$

where $\rho^{\pi}_{\text{re}}(\check{æ}_{<t}) := \sum_{\pi_{2:t}} \rho^{\pi}_{\text{re}}(æ'_{<t}\pi_{2:t}) = \sum_{\pi_{2:t}} \rho^{\pi}_{\text{re}}(æ_{<t})$.

The value-equivalence follows from that the realistic value functions measure ρ^{π}_{re}-expected u_1-utility, and the standard value functions measure $\rho^{\check{\pi}}$-expected u_1-utility:

$$\begin{aligned} Q^{\text{re}}_1(\epsilon\pi(\epsilon)) &= \mathbb{E}_{\check{æ}_{1:\infty} \sim \rho^{\pi}_{\text{re}}}\left[\sum_{k=1}^{\infty} \gamma^{k-1} u_1(\check{æ}_{<k})\right] \\ &= \mathbb{E}_{\check{æ}_{1:\infty} \sim \rho^{\check{\pi}}}\left[\sum_{k=1}^{\infty} \gamma^{k-1} u_1(\check{æ}_{<k})\right] = Q^{\check{\pi}}(\epsilon\check{\pi}(\epsilon)). \end{aligned} \qquad \square$$

Optimal policies. We are now ready to show that an optimal policy exists. We treat two cases: Utility modification and policy modification. In the utility modification case, we only need to show that an optimal policy exists. In the policy modification case, we also need to show that we can add a name for the optimal policy. The idea in both cases is to build from an optimal world policy $\check{\pi}^*$, and use that associated policies have the same value by Lemma 19.

In the utility modification case, the policy names $\mathcal{P}$ are the same as the utility functions $\mathcal{U}$, with $\iota(u) = \pi^*_u = \arg\max_{\pi} Q^{\text{re},\pi}_u$. For the utility modification case, it therefore suffices to show that an optimal policy π^*_u exists for arbitrary utility function $u \in \mathcal{U}$. If π^*_u exists, then u is a name for π^*_u; if π^*_u does not exist, then the naming scheme ι is ill-defined.

Theorem 20 (Optimal policy existence, utility modification case)**.** *For any modification-independent utility function u_t, there exists a modification-independent, non-modifying policy π^* that is optimal with respect to V^{re}_t.*

Proof. By the compactness argument of Lattimore and Hutter (2014, Thm. 10) an optimal policy over world actions $(\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$ exists. Let $\check{\pi}^*$ denote such a policy, and let $\pi^*(h) = (\check{\pi}^*(\check{h}), \pi^*)$. Then π^* is a non-modifying optimal policy. Since any policy has realistic value corresponding to its associated world policy by Lemma 19 and the associated policy of π^* is $\check{\pi}^*$, it follows that π^* must be optimal. □

For the policy-modification case, we also need to know that the optimal policy has a name. The naming issue is slightly subtle, since by introducing an extra name for a policy, we change the action space. The following theorem shows that we can always add a name p^* for an optimal policy. In particular, p^* refers to a policy that is optimal in the extended action space $\mathcal{A}' = \check{\mathcal{A}} \times (P \cup \{p^*\})$ with the added name p^*.

Theorem 21 (Optimal policy name)**.** *For any policy-modification model* $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}, \iota)$ *and modification independent belief and utility function* ρ *and* u*, there exists extensions* $\mathcal{P}' \supseteq \mathcal{P}$ *and* $\iota' \supseteq \iota$*,* $\iota' : \mathcal{P}' \to \Pi$*, such that an optimal policy* π^* *for* $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}', \iota')$ *has a name* $p^* \in \mathcal{P}'$*, i.e.* $\pi^* = \iota'(p^*)$*. Further, the optimal named policy* π^* *can be assumed modification-independent and non-modifying.*

Proof. Let $\check{\pi}^*$ be a world policy $(\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$ that is optimal with respect to the standard value function V (such a policy exists by Lattimore and Hutter (2014, Thm. 10)).

Let p^* be a new name $p^* \notin \mathcal{P}$, $\mathcal{P}' = \mathcal{P} \cup \{p^*\}$, and define the policy $\pi^* : (\check{\mathcal{A}} \times \mathcal{P}' \times \mathcal{E})^* \to (\check{\mathcal{A}} \times \mathcal{P}')$ by $\pi^*(h) := (\check{\pi}^*(\check{h}), p^*)$ for any history h. Finally, define the extension ι' of ι by

$$\iota'(p) = \begin{cases} \iota(p) & \text{if } p \in \mathcal{P} \\ \pi^* & \text{if } p = p^*. \end{cases}$$

It remains to argue that π^* is optimal. The associated world policy of π^* is $\check{\pi}^*$, since π^* is non-modifying and always takes the same world action as $\check{\pi}^*$. By Lemma 19, all policies for $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}', \iota')$ have values equal to the value of their associated world policies $(\check{\mathcal{A}} \times \mathcal{E})^* \to \check{\mathcal{A}}$. So π^* must be optimal for $(\check{\mathcal{A}}, \mathcal{E}, \mathcal{P}', \iota')$ since it is associated with an optimal world policy $\check{\pi}^*$. □