



# Affinity-Based Reinforcement Learning: A New Paradigm for Agent Interpretability

---

Charl Maree

---



Charl Maree

# Affinity-Based Reinforcement Learning: A New Paradigm for Agent Interpretability

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at  
the Faculty of Engineering and Science, Specialisation in Information and  
Communication Technology

University of Agder  
Faculty of Engineering and Science  
2023

Doctoral Dissertations at the University of Agder 395

ISSN: 1504-9272

ISBN: 978-82-8427-108-8

© Charl Maree, 2023

Printed by Aksell

Kristiansand, Norway

# Abstract

The steady increase in complexity of reinforcement learning (RL) algorithms is accompanied by a corresponding increase in opacity that obfuscates insights into their devised strategies. Methods in explainable artificial intelligence seek to mitigate this opacity by either creating transparent algorithms or extracting explanations post hoc. A third category exists that allows the developer to affect what agents learn: constrained RL has been used in safety-critical applications and prohibits agents from visiting certain states; preference-based RL agents have been used in robotics applications and learn state-action preferences instead of traditional reward functions. We propose a new affinity-based RL paradigm in which agents learn strategies that are partially decoupled from reward functions. Unlike entropy regularisation, we regularise the objective function with a *distinct* action distribution that represents a desired behaviour; we encourage the agent to act according to a prior while learning to maximise rewards. The result is an inherently interpretable agent that solves problems with an intrinsic affinity for certain actions. We demonstrate the utility of our method in a financial application: we learn continuous time-variant compositions of prototypical policies, each interpretable by its action affinities, that are globally interpretable according to customers' financial personalities.

Our method combines advantages from both constrained RL and preference-based RL: it retains the reward function but generalises the policy to match a defined behaviour, thus avoiding problems such as reward shaping and hacking. Unlike Boolean task composition, our method is a fuzzy superposition of different prototypical strategies to arrive at a more complex, yet interpretable, strategy.

# Sammendrag

Kompleksitetsnivået til algoritmer innenfor forsterkende læring (RL) øker stadig slik at metodikkens transparens blir redusert, noe som hindrer vår innsikt inn i de lærte strategiene. Derfor har det blitt utviklet forskjellige metoder innen forklarbar kunstig intelligens som sikter på å minske denne ugjennomsiktigheten, enten ved bruk av transparente algoritmer eller utdrag av forklaringer etter læringen. Det eksisterer en tredje kategori som gjør det mulig for utviklere å påvirke hva RL agentene lærer: Begrenset RL har blitt brukt i sikkerhetskritiske applikasjoner og forhindrer at agentene havner i visse tilstand, mens preferansebaserte RL agenter har blitt brukt i robotapplikasjoner og lærer preferanser for visse stater og aksjoner istedenfor tradisjonelle belønningsfunksjoner. Vi foreslår et nytt affinitetsbasert RL paradigme der agentene lærer strategier som er delvis frikoblet fra belønningsfunksjoner. I motsetning til entropiregularisering regulariserer vi objektivfunksjonen med en unik handlingsfordeling som representerer en ønsket atferd. Vi oppfordrer agenten til å handle i henhold til en forutsetning mens den lærer å maksimere belønningene. Resultatet er en iboende tolkbar agent som løser problemer med en affinitet for visse handlinger. Vi demonstrerer nytten av metoden vår i en anvendelse innenfor finans: Vi lærer kontinuerlige tidsvariante sammensetninger av prototypiske retningslinjer, hver tolkbar via sine handlingstilhørigheter, som er globalt tolkbare i henhold til kundenes økonomiske personligheter.

Vår metode kombinerer fordelene fra både begrenset RL og preferansebasert RL: Vi beholder belønningsfunksjonen, men generaliserer policyen for å matche en definert atferd, og unngår dermed problemer som belønningsforming og hacking. I motsetning til Boole'sk oppgavesammensetning er metoden vår en superposisjon av forskjellige prototypiske strategier som gir en mer kompleks, men likevel tolkbar, strategi.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Ideation of Affinity-Based RL . . . . .	1
1.2	Explainability vs. Interpretability . . . . .	1
1.3	Explainable AI in Financial Technology . . . . .	2
1.4	Ambition, Desiderata, and Challenges . . . . .	3
1.5	Generic Applicability . . . . .	5
1.6	Research Design . . . . .	5
1.7	Contributions . . . . .	6
1.8	Thesis Outline . . . . .	7
<b>2</b>	<b>Reinforcement Learning, Explainability, and Interpretability</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Regularisation, Exploration, and Exploitation . . . . .	10
2.3	Constrained Reinforcement Learning . . . . .	11
2.4	Preference-Based Reinforcement Learning . . . . .	12
2.5	Explainable and Interpretable Reinforcement Learning . . . . .	13
2.5.1	Deductive Explanations . . . . .	14
2.5.2	Inductive Explanations . . . . .	19
2.5.3	Summary . . . . .	23
2.6	Potential Research Opportunities . . . . .	23
<b>3</b>	<b>Principles of Affinity-Based Reinforcement Learning</b>	<b>27</b>
3.1	Conceptual Formulation . . . . .	27
3.2	An Illustrative Example of Affinity-Based RL . . . . .	28
3.3	Agent Prototyping with Affinity-Based RL . . . . .	29
3.4	Composition of Traits and Policies . . . . .	31
<b>4</b>	<b>Empirical Methodology</b>	<b>33</b>
4.1	Empirical Justification for Affinity-Based RL . . . . .	33
4.2	Application of Affinity-Based RL in Finance . . . . .	34

4.2.1	Application Overview . . . . .	34
4.2.2	Feature Extraction with RNN . . . . .	34
4.2.3	Affinity-Based RL for Personal Investment Advice . . . . .	36
4.2.4	Agent Composition using RNN . . . . .	38
<b>5</b>	<b>Application in Financial Advising</b>	<b>39</b>
5.1	Explainable Transaction Classification . . . . .	39
5.2	Customer Micro-Segmentation . . . . .	40
5.3	Affinity-Based RL for Investment Advice . . . . .	42
5.3.1	Composition of Prototypical Investment Agents . . . . .	44
5.3.2	Time-Variant Composition . . . . .	47
5.3.3	Explaining Prototypical Agents with Markov Models . . . . .	49
5.4	Discussion . . . . .	50
<b>6</b>	<b>Conclusions and Future Research Directions</b>	<b>53</b>
	<b>Bibliography</b>	<b>55</b>
	<b>Appended Papers</b>	<b>63</b>
<b>A</b>	<b>Towards Responsible AI for Financial Transactions</b>	<b>63</b>
<b>B</b>	<b>Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality</b>	<b>71</b>
<b>C</b>	<b>Understanding Spending Behavior: Recurrent Neural Network Ex- planation and Interpretation</b>	<b>77</b>
<b>D</b>	<b>Balancing Profit, Risk, and Sustainability for Portfolio Manage- ment</b>	<b>85</b>
<b>E</b>	<b>Reinforcement Learning Your Way: Agent Characterization through Policy Regularization</b>	<b>95</b>
<b>F</b>	<b>Can Interpretable Reinforcement Learning Manage Prosperity Your Way?</b>	<b>107</b>
<b>G</b>	<b>Reinforcement Learning with Intrinsic Affinity for Personalized Prosperity Management</b>	<b>121</b>
<b>H</b>	<b>Symbolic Explanation of Affinity-Based Reinforcement Learning Agents with Markov Models</b>	<b>145</b>



<b>I Towards Artificial Virtuous Agents: Games, Dilemmas and Machine Learning</b>	<b>163</b>
---	------------

# Chapter 1

## Introduction

### 1.1 Ideation of Affinity-Based RL

Recent advances in artificial intelligence (AI) have introduced a complexity that obfuscates the intricacies of what models learn, thus hindering our understanding of model behaviour [1]. Reinforcement learning (RL) is not immune to this phenomenon, and its complexity poses particular challenges related to explainability and interpretability [2, 3, 4]. Furthermore, shaping objective functions for complex tasks is notoriously difficult as developers strive to control what agents learn [5]; the added burden of preferring desired, or avoiding undesired, behaviours has been detrimental to the utility—and convergence—of objective functions and has led to alternative approaches such as constrained RL and preference-based RL [5, 6]. We instead impose desired behaviours, and avoid undesired ones, in a parsimonious and transparent process. We propose a new paradigm in RL, which we refer to as affinity-based RL (ab-RL), that decomposes strategies into elemental policies with defined—and interpretable—desired and / or undesired behaviours. We use regularisation to imprint these behaviours into the policies of a set of prototypical<sup>1</sup> agents, which we then compose to form superpositions that solve the original problem in an interpretable way without added complexities that may inundate the objective function.

### 1.2 Explainability vs. Interpretability

An explanation of an agent’s behaviour does not automatically and necessarily lead to an interpretation, i.e. a description—let alone guarantee—of an overall strategy

---

<sup>1</sup>We refer to the following definition of “prototypical”: an instance or entity that illustrates the typical qualities of a class or group. The word prototype comes from the Latin words *proto*, which means original, and *typus*, which means form or model. A prototype is an especially representative example of a given category.

or desired behaviour. Nevertheless, the terms “explainability” and “interpretability” are often used interchangeably, thus hindering the establishment of a consensus on the scope and definition of AI explainability [1, 7]. Barredo Arrieta et al. [1] proposed the following distinction: *explainability* refers to “an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans”, and *interpretability* refers to “the ability to explain or to provide the meaning in understandable terms to a human”. This definition of explainability, which involves human understanding, necessitates the concept of a target audience, i.e. model developers, domain experts, end users, regulatory authorities, executive managers, etc. We argue that this is a gratuitous emphasis on human understanding—the definition of interpretability, after all, mainly entails human understanding—that contributes to a lack of metrics and measures that quantify and compare the explainability of AI systems [7]. Therefore, to generalise AI explainability and facilitate direct comparison between different explanations, we eliminate this need for a target audience; we simply define *explainability* as a symbolic representation of a model’s predictions, and *interpretability* as the tools needed for humans to reason about a model’s behaviour.

### 1.3 Explainable AI in Financial Technology

Modern financial services are driven by an increase in demand for personalisation, and as customer bases grow, the need for automation has proliferated AI into a ubiquitous tool in the sector. One example is robo-advising: human investors commonly deviate from the optimum strategy of a fully rational agent and their decisions can be significantly improved by avoiding typical indiscretions such as low participation, poor diversification, default bias, portfolio inertia, excessive trading, and trend chasing [8]. While these can be mitigated through expert investment advice, human advisors are not devoid of bias and professionally managed portfolios are often more closely correlated to the preferences of the portfolio manager than those of the investor [8]. Robo-advisors offer tailored strategies for allocating funds across multiple asset classes which involve two crucial and continuous phases: individual profiling and asset allocation. While such algorithms are typically opaque, financial service providers are subject to fiduciary duty. Therefore, recent efforts seek transparency through the use of explainable AI methods in areas such as customer profiling [9], investment and portfolio management [10], lending and credit management [11], fraud detection and anti-money laundering [12], and auditing [13]. Beyond explainability, such endeavours have not only resulted in significant model improvements [10],

but also in revealing new research hypotheses that would otherwise have remained obscure [9].

## 1.4 Ambition, Desiderata, and Challenges

Our ambition was to develop an AI-powered system that (1) gives financial investment advice based on customer profiles that may change over time, (2) learns optimal investment strategies using RL, (3) assumes that personalised strategies will be an amalgam of basic financial behaviours and insights, and (4) makes the learnt strategy accessible for explanation and interpretation. Motivated by this ambition, we developed a generic affinity-based RL paradigm that embodies the basic desired traits that impact a strategy and that may fluctuate over time. It represents the amalgam of strategies to be learnt as a fuzzy<sup>2</sup> superposition of basic prototypical strategies; the final strategy becomes a superposition of the prototypical trajectories in the state-action space. Unlike previous applications of hierarchical RL that sequentially orchestrated specific agents' strategies, ab-RL composes such strategies into superpositions that are interpretable as the weighted average of their parts.

The key desiderata of our ambition elicit four research questions. These were the drivers of our research, and each application area revealed a generic element of the ab-RL framework. Customer profiling, firstly, is a nontrivial endeavour and traditional methods involve features such as age, gender, ethnicity, and postal codes, that render individuals susceptible to discrimination [14]. Our challenge was to use a fair metric, i.e. financial transactions, to classify customers according to their spending behaviour.

**Research Question 1** *How can we distinguish between agent profiles based on their behaviour that may change over time?*

Task decomposition, for the purpose of interpretability, must involve domain-specific traits. These traits deconstruct the natural behaviour of a rational agent into elemental parts. Based on this behavioural classification, our second challenge was to reveal those characteristic traits inherent to customer spending behaviour. Our digital footprints, of which our financial transactions are manifestations, are predictors of our personality traits [9]. The Big Five personality model presents a set of traits that represent behavioural patterns across cultures, nationalities, and languages [9]. We used these five personality traits to define the desired behaviours of five proto-

---

<sup>2</sup>The fuzzy superposition is a weighted composition of the prototypical policies which implement desired behaviours. These weights are in the range [0, 1].

typical agents; each agent is associated with one of the five personality traits and has an inherent affinity for certain investment types.

**Research Question 2** *How can we model an actor’s sequence of decisions based on certain domain-specific traits?*

Each behavioural trait uniquely associates with each of the agent actions. These associations describe the expected behaviour of a prototypical agent that perfectly represents a given trait. The complete probabilistic set of action associations characterises the prototypical agent’s action affinity. We trained five prototypical agents with intrinsic affinities for different types of investment assets. Each agent represented a specific personality trait and invested only in those asset types that are associated with that personality trait.

**Research Question 3** *How can we design RL agents that exhibit locally-optimal and desired behaviours?*

Prototypical agents learn specific strategies, i.e. their choices of actions must follow a deterministic and desirable distribution. Current methods such as constrained RL—that prevents certain states—and preference-based RL—that requires the arduous definition of expert preferences per state, without the use of a reward function—are insufficient for our purpose. Our aim was to instil both desired and undesired preferences, while retaining a basic reward function, i.e. the maximisation of profit. Policy regularisation guarantees convergence at a local optimum, but current approaches merely aim to improve convergence, for example, by encouraging general exploration [15, 16]. Our affinity-based regularisation encourages exploration of a *specific* region of the state-action space, thus ensuring a local optimum solution that closely mimics the defined action distribution. Finally, spending behaviour naturally fluctuates over time, and our investment advice must respond in accordance with these changes. However, basic personality traits and their associations with different investment classes remain consistent over time.

**Research Question 4** *How can we compose multiple prototypical strategies in a time-variant way to solve problems according to preferences that may change over time?*

The advantage of a prototypical decomposition goes beyond interpretability; it enhances simplicity. Without additional learning, we may combine these prototypical strategies in different ratios to represent new strategies. The relative contributions of each prototypical policy that form a superposition can be modelled in a time-variant way, e.g. with recurrent neural networks.

## 1.5 Generic Applicability

Affinity-based RL is a generically applicable paradigm; similar to entropy-based RL, its objective function is the sum of expected cumulative rewards and a regularisation term. The original aim of policy regularisation was to improve convergence and it is guaranteed to have no adverse effects [15, 16]. In essence, its underlying and unstated intention is to assert influence over the learning process. Affinity-based RL controls the learning process by using a specific probabilistic action distribution as the regularisation term. This shifts the exploration / exploitation balance such that the policy observes an overall action probability distribution; it instils an intrinsic affinity for certain actions while discouraging the choice of others. This concept can be applied to a wide range of problems, i.e. in any circumstance where there is a need to assert control over the learning process for the purpose of interpretability, convergence, trust, etc. Affinity-based RL is currently being investigated—and potentially extended—for applications such as modelling climate change interventions as social dilemmas, personalised learning and teaching based on student profiles, control of wind farms which balances power production and remaining useful life, chronic disease treatment based on individual situations, and virtuous agents that learn to deal with moral dilemmas [17].

## 1.6 Research Design

Although our research was driven by a specific application, i.e. personal investment advice, we designed a generic framework for our methodology. Figure 1.1 illustrates this proposed framework.

The process of decomposing and recomposing tasks might seem circuitous, but it evades complexities, such as reward shaping, associated with personalisation of strategies. In addition, it eliminates the need to retrain agents when unique individuals join the solution. Such new individuals simply require a new composition of the pre-trained prototypical agents. Finally, the guarantees that follow from policy regularisation ensure the interpretability of the prototypical agents and, as a linear combination of its parts, the composed strategy. Therefore, it is unlikely that, for sufficiently complicated applications, a solution exists where an RL agent could directly learn such interpretable and personalised strategies.

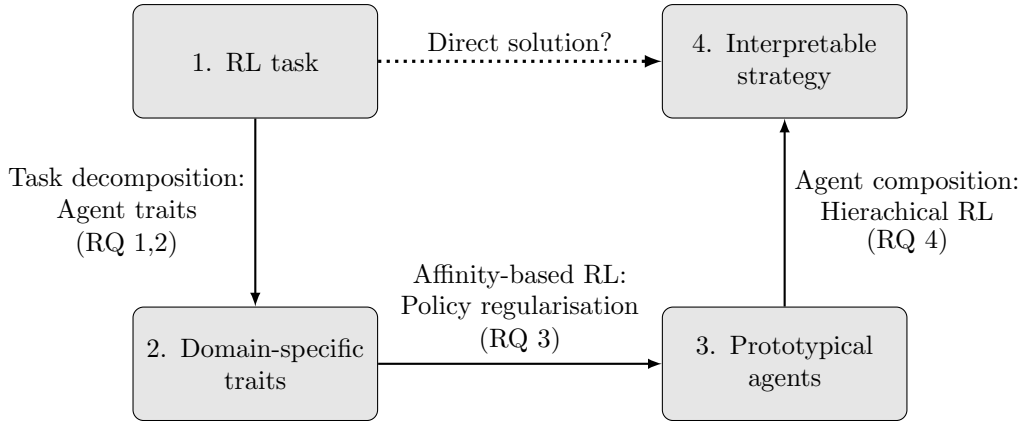


Figure 1.1: An illustration of the generic ab-RL framework. It decomposes tasks into elemental, domain-specific, traits. A set of prototypical agents are trained with affinities that associate them with these traits. These agents are finally combined to form an amalgamated superposition of their strategies that reflect individual portrayals of the traits; each individual or entity portrays a unique combination of the traits, and their personal strategy is matched to this unique characterisation. It is not certain that a direct solution exists where interpretable agents learn to solve complicated, personalised tasks, without the need for retraining when new individuals or entities join.

## 1.7 Contributions

We present the generic ab-RL framework, i.e. a new paradigm in RL that guarantees locally-optimum solutions that adhere to predefined action distributions and are partially decoupled from the reward function. Unlike constrained RL, it makes no distinction between desired and undesired behaviour, and unlike hierarchical RL it blends elemental strategies to form fuzzy non-sequential superpositions. Similar to hierarchical RL, however, ab-RL enhances interpretability through the decomposition of tasks.

Our value proposition, and second major contribution, is a financial application of this framework that recommends personal investment strategies based on individual spending behaviours, as demonstrated in customers’ financial transactions. Our preliminary steps, i.e. customer profiling or micro-segmentation along personality traits, are therefore necessary to illustrate the paradigm of decomposing and recomposing prototypical strategies. We presented this application to a major Norwegian bank where it was well received and who are interested in implementing it as a new product.

## 1.8 Thesis Outline

Chapter 2 gives an overview of relevant current methodology and presents a taxonomy of the state of the art in explainable and interpretable reinforcement learning. We conceptualise our ab-RL framework in Chapter 3 and detail our empirical methodology in Chapter 4. Chapter 5 presents the results of our financial application and Chapter 6 presents our conclusions and proposes directions for future research.





# Chapter 2

## Reinforcement Learning, Explainability, and Interpretability

### 2.1 Introduction

In reinforcement learning (RL), agents learn to solve problems through trial and error while maximising reward expectations. These agents typically interact with their environments in discrete time steps; at each time step  $t$ , an agent performs an action based on the current state  $s_t$ , for which it receives a reward  $r_{t+1}$  and a new state  $s_{t+1}$ . We illustrate this general RL framework in Figure 2.1.

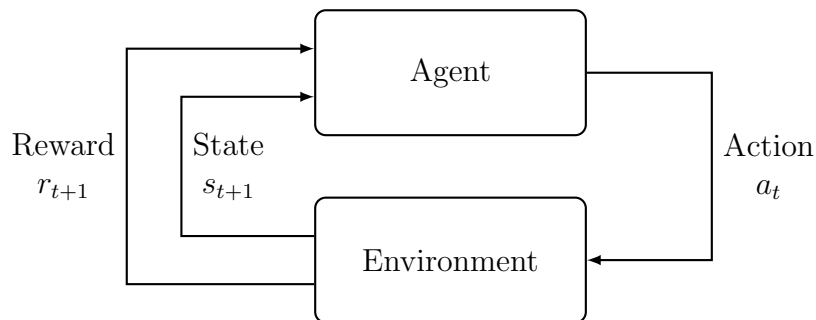


Figure 2.1: An information flow diagram illustrating the typical RL framework. An agent performs a series of discrete-time actions  $a_t$  in an environment that invariably responds with an immediate reward  $r_{t+1}$  and a new state  $s_{t+1}$ .

The environment is typically modelled as a discrete-time Markov decision process (MDP) in which the next state depends only on the current state and the action taken by the agent. An MDP is described by the tuple  $(S, A, R, P)$  where  $S$  is the set of states,  $A$  the set of actions,  $R(s, a)$  the reward for action  $a \in A$  in state  $s \in S$ , and  $P(s, a) = P(s'|s, a)$  the probability of transitioning from state  $s$  to state  $s'$  as a result of action  $a$  [18].

The agent's goal is to learn the policy  $\pi(s) = P(a|s)$  that maximises the expected

cumulative reward:

$$\mathbb{E} [R(s, a)] = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]$$

where  $\gamma \in [0, 1]$  is a discount factor and  $r_t$  is the immediate reward at the time step  $t$  [19].

There exist different classes of learning algorithms, i.e. deep Q-learning that is restricted to discrete, and finite, state-action spaces, and policy gradient algorithms that are suitable for continuous, therefore infinite, state-action spaces. There are numerous advantages to policy gradient methods, most notable is that the state representations can be chosen such that they are meaningful to the task. In general, policy gradient methods optimise the objective function.

$$J(\theta) = \mathbb{E} \left[ \sum_{t=0}^T \gamma r_t \right]$$

where  $\theta$  represents the model parameters. These model parameters are updated according to the gradient update rule following the steepest descent of the expected return:  $\theta_{k+1} = \theta_k + \alpha \nabla J(\theta_k)$  where  $\alpha \in \mathbb{R}^+$  is the learning rate and  $k \in \mathbb{Z}^+$  is the current update iteration. Deep deterministic policy gradients (DDPG) is a model-free policy gradient algorithm that consists of four artificial neural networks: an actor  $\mu(\theta_\mu)$  representing the policy, a critic  $Q(\theta_Q)$  representing the state action value function, and for numerical stability, a target actor  $\mu'(\theta'_\mu)$  and a target critic  $Q'(\theta'_Q)$  [20]. During learning, the target network parameters are slowly updated, given a soft update parameter  $\tau \in [0, 1]$  that typically has a small value:  $\theta'_i = \tau \theta_i + (1 - \tau) \theta'_i$ ,  $i \in \{\mu, Q\}$ . DDPG has been used successfully in many applications, including robotic navigation tasks [21], energy-aimed train timetable rescheduling [22], and stock portfolio optimization [23, 24].

In this chapter, we recount RL regularisation methods, give background into constrained and preference-based RL, present a brief survey of the state of the art in explainable RL, and highlight potential research opportunities.

## 2.2 Regularisation, Exploration, and Exploitation

While RL is particularly adept at learning in the presence of sparse and delayed rewards, it is often plagued by a trade-off between exploiting known good solutions or exploring for better unknown solutions; this is known as the exploration / exploitation dilemma [19]. The  $\epsilon$ -greedy strategy aims to promote exploration by selecting the assumed best action most of the time, while acting randomly with

a small probability  $0 < \epsilon \ll 1$  [19]. Intrinsic motivation is a more sophisticated approach that enables agents to learn strategies that are partially decoupled from the expected rewards [25, 26]. One such method is entropy regularisation; it has been proven to improve learning performance, while never being detrimental to convergence [15, 27]. Entropy regularisation adds a distance-based penalty  $\mathcal{H}(\pi, \pi_0)$ , scaled by a hyperparameter  $\lambda$ , to the objective function.

$$J(\theta) = \mathbb{E}[R(s, a)] - \lambda \mathcal{H}(\pi, \pi_0)$$

To maximise the entropy of the policy  $\pi$  during learning, the prior  $\pi_0$  is a global uniform action distribution; the action distribution is uniform across all states, which encourages exploration independent of both the state and reward function. Galashov et al. [16] generalises this approach with a regularisation term proportional to the Kullback-Leibler (KL) divergence between the state-action distribution of the policy and that of a given prior:  $\mathcal{H}_{KL}(\pi, \pi_0) = \sum_{s \in \mathcal{S}} \pi(s) \log \left( \frac{\pi(s)}{\pi_0(s)} \right)$ . It uses a local action distribution that encourages a state-dependent behaviour with the explicit purpose of improved learning convergence [16]

## 2.3 Constrained Reinforcement Learning

In contrast to policy regularisation, which *promotes* desired behaviours, constrained RL *avoids* undesired conditions. In their most simple form, these conditions are enforced by altering the value function:

$$\begin{aligned} \pi(s) &= \operatorname{argmax}_{a'} Q(s, a') \\ a' &= a + \xi_\phi(s, a, \Phi) \end{aligned}$$

where  $\xi_\phi(s, a, \Phi)$  is a perturbation function that alters action  $a$  in state  $s$  with a value in the range  $[-\Phi, \Phi]$ . Chow et al. [28] extended this approach to policy gradient methods; they proposed a method that penalises the value function with the accumulated cost of a series of actions, with the intention of avoiding specific actions in specific states. They define a cumulative risk value for events that may occur with small probability but with high consequence. This risk-related cost of a state-action pair  $\mathcal{G}(s, a)$  is the sum of discounted costs  $C(s_k, a_k)$  incurred when following the current policy  $\pi(s)$ , i.e.  $\mathcal{G}(s, a) = \sum_{k=0}^T \gamma C(x_k, a_k) | \pi(s)$ .

However, defining the costs for each state-action pair can be an arduous process, especially for large state-action spaces. Qin et al. [29] address this limitation by

imposing constraints on state density functions. State density is related to the state distribution and is a measure of how often a state is visited; it is the discounted sum of probabilities of visiting state  $s$  at time  $t$ , formally:  $\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma P(s_t = s | \pi, s_0)$ . These state densities affect the objective function to be maximised during learning:

$$J(\theta) = \int_S \int_A \rho_\pi(s, a) r(s, a) da ds$$

$$s.t. \quad \rho_{min}(s) < \rho_\pi(s) < \rho_{max}(s)$$

Miryoosefi et al. [6] proposed a method that penalises the reward function with the Euclidean distance between the current state and a given set of restrictions. They define a long-term loss  $Z$  as the expected sum of discounted losses:

$$Z(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma z_t | \pi \right]$$

$$z = \min_{\pi \in \Pi} [dist(s, \mathcal{C}) | \pi]$$

where  $\Pi$  is a finite set of candidate policies,  $dist$  represents the Euclidean distance between a point and a set of points, and  $\mathcal{C}$  is the defined set of undesirable states.

While methods that constrain the state space are useful in safety-critical applications, they do little to aid in interpretability of policies. Granted, they provide assurances that certain states will not be visited or that certain actions will not be taken in certain states, but this is insufficient to characterise a policy. Instead, information about both desired and undesired strategies provide the required insight into the natural behaviour of agents.

## 2.4 Preference-Based Reinforcement Learning

Designing reward functions often requires both comprehensive domain knowledge and an intricate understanding of the selected RL algorithm; the developer must consider both the desired behaviour to be learnt and the learning process itself [5]. Preference-based RL aims to mitigate these problems; it learns directly from domain experts' state-dependent action preferences, instead of a reward function. By eliminating the reward function, it addresses issues such as reward shaping—the addition of supplemental rewards to encourage a desired learning objective—and reward hacking—a type of overfitting, where agents may learn the literal reward function instead of generalising its intent. Fürnkranz and Hüllermeier [30] define several preference-based relationships between certain defined choices  $z_i \in Z$ :

- $z_i \prec z_j$ :  $z_i$  is strictly preferred over  $z_j$
- $z_i \preceq z_j$ :  $z_i$  is weakly preferred over  $z_j$
- $z_i \sim z_j$ :  $z_i$  and  $z_j$  are equally preferable

In the RL context, training data are given in the form of pairwise comparisons between actions that are associated with states. Similar to action-value functions, states may then be ranked according to desirability by ranking the preferences of actions in each state. The most substantial problem with preference-based RL is that it requires a large set of preferences, or particularly well-defined preference functions in the case of continuous—and infinite—state-action spaces [5]. Defining these preferences can be tedious, and the problem of complicated reward functions may merely be transferred to complicated preference functions. We propose an affinity-based RL approach that does not suffer from this elaboration of the utility function; it maintains the traditional reward function but uses regularisation to guide the learning process, and thus instils an intrinsic action affinity.

## 2.5 Explainable and Interpretable Reinforcement Learning

Human reasoning is generally classified into two broad categories: deductive and inductive reasoning [31]. While deductive reasoning is the drawing of logical conclusions based on known facts, inductive reasoning is the generalisation of observations, which may be incomplete samples of reality [31]. We created a taxonomy of explainable<sup>1</sup> RL on this classification of human reasoning. Specifically, we construe that *deductive explanations* are logical explanations based on premises drawn from statistical or systematic analyses of the policy, while *inductive explanations* are conclusions drawn from observations and, therefore, are generalisations of the policy. Deductive explanations can be subclassified into two subclasses: *explanation and policy integration* where the explanations are conjugate to the policy, e.g. through direct integration into the value function, and *hierarchisation and decomposition* where the states, actions, or rewards are decomposed such that intrinsically explainable policies may be learnt. Figure 2.2 is an illustration of our taxonomy and lists the different approaches discussed here. We limit our scope to approaches that specifically explain deep RL and we ignore alternative methods that *replace* deep RL

---

<sup>1</sup>The lack of consensus on the definitions of explainability and interpretability results in most authors using these terms interchangeably. For brevity, we simply use the term explainability when referring to explainability / interpretability in our taxonomy.

with some other intrinsically explainable system, such as Programmatically Interpretable Reinforcement Learning (PIRL) [32], Linear Model U-Trees (LMUT) [33], etc.

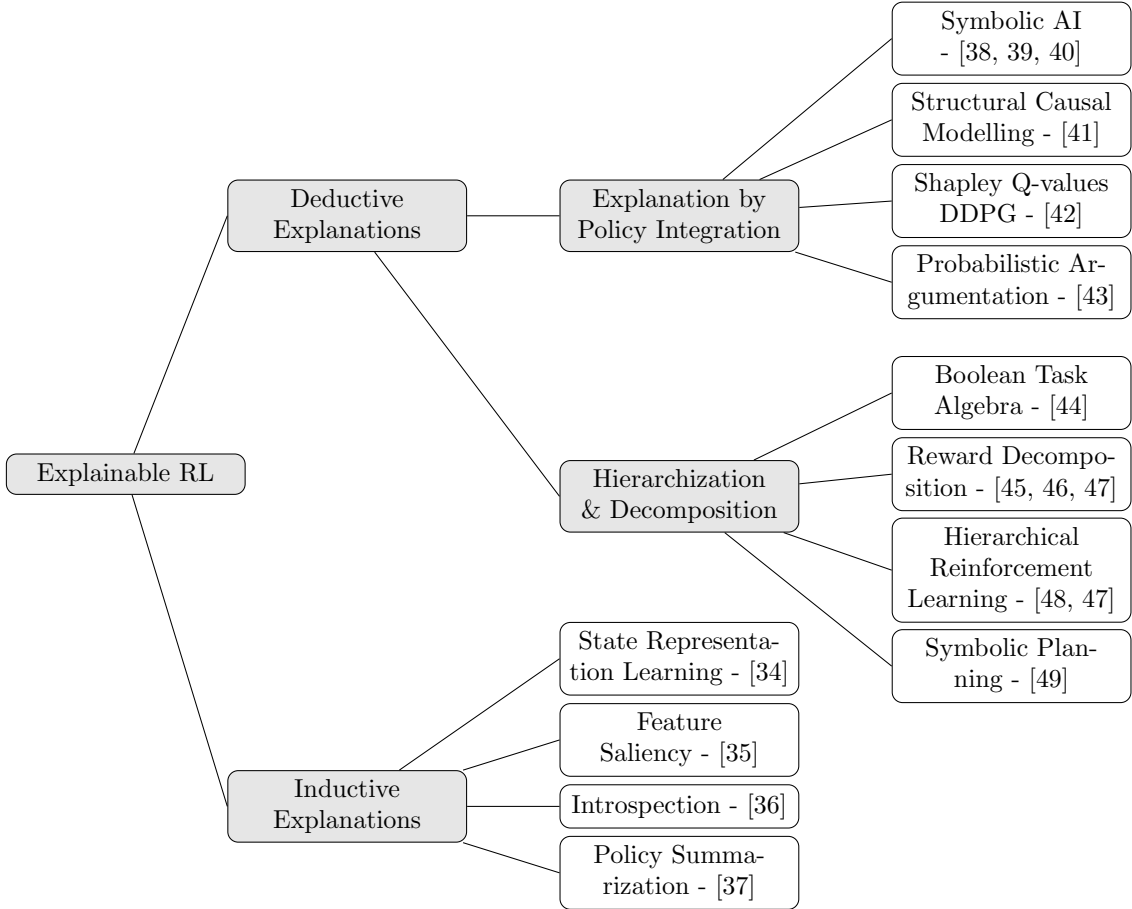


Figure 2.2: A taxonomical classification of different methods used in explainable reinforcement learning. Gray shaded rectangles represent classes, while clear rectangles represent methods.

### 2.5.1 Deductive Explanations

Deductive explanations are highly integrated with the policies they explain, which brings forth certain advantages such as the potential for very high fidelity, but they also inherit disadvantages from the environments for which they were designed, e.g. some are limited to discrete state-action spaces. This class of methods can be subdivided into two subclasses: *Explanation by policy integration* and *hierarchisation and decomposition*. Methods in the class *explanation by policy integration* learn explanations either during or directly after policies have been learnt. These explanations are therefore closely coupled to their policy, which potentially improves fidelity, but often at the expense of complexity. The non-expert interpretability of some of these methods may therefore be insufficient. We review four methods in this

subclass: symbolic AI, structural causal modelling (SCM), Shapley Q-values deep deterministic policy gradients (SQDDPG), and probabilistic argumentation (PA).

Explainability using symbolic AI facilitates the use of domain knowledge in simplifying the state-action space such that humans may reason about the policy. Garnelo et al. [38] developed a system that uses a deep learning back-end to encode the state space into usable features, which a human expert then combines into symbolic representations. These symbolic representations are then used to learn an inherently explainable policy. Garnelo et al. [38] illustrated the efficacy of their system through a toy problem in which an agent learnt to navigate a grid in search of objects with a given shape while avoiding other objects. Interactions between the agent and the objects resulted in rewards and altered object shapes. They used a convolutional neural network (CNN) to encode the state—pixels in the grid—and used these extracted features to build new *symbolic* states. These symbolic states were sets of tuples  $s = \{(t_i, t'_i, d_i), i \in [1, N]\}$  where  $N$  is the number of objects in the grid,  $t_i$  is the named shape of object  $i$  *before* the action,  $t'_i$  is the named shape of object  $i$  *after* the action, and  $d$  is the two-dimensional change in the agent’s location due to the action. These symbolic states were used in the conventional RL setting:  $(s, s', a, r)$  where  $s$  represents the current symbolic state,  $s'$  the next symbolic state,  $a$  the action, and  $r$  the reward. According to the authors, the simplified symbolic states in  $s$  facilitated reasoning about the policy in terms of object types and agent locations, as opposed to pixels in a grid which would have been the conventional representation of the state.

Structural causal modelling (SCM) aims to mimic the type of explanations that occur naturally in human reasoning: cause-and-effect relations (“Why action X?”) and counterfactual explanations (“Why not action Y?”) [41]. SCM learns *causal relationships* between states, actions, and rewards by defining *action influence graphs* that describe the causal effects of applying actions in states. These action influence graphs depict all possible paths from an originating state (head node), using all possible actions and state transitions, to all reachable pseudoterminal states (sink nodes), given a predefined set of sink nodes. Then, using the learnt policy, a *causal chain* is defined as the single path in the action influence graph that links the head node to a specific sink node determined by the policy. A *complete explanation* for an action is given by the tuple  $(X_r, X_h, X_i)$  where  $X_r$  is the vector of rewards reached by following the causal chain,  $X_h$  is the vector of features that describe the head node, and  $X_i$  is the vector of features that describe all intermediate nodes. Recognising that this explanation could become unintelligible due to a potentially large number of intermediate nodes, Madumal et al. [41] also define a *minimal complete explana-*



tion, defined by the tuple  $(X_r, X_h, X_p)$  where  $X_r$  and  $X_h$  are unchanged from the complete explanation and  $X_p$  are the vectors of features of nodes that are immediate predecessors of any node in  $X_r$ , i.e. all states that resulted in a reward. SCM generates *counterfactual explanations* by comparing the causal chains, as determined by the policy, under the following conditions: a) the current conditions under which the current action was chosen and b) a *counterfactual instantiation* where the features for the head node are perturbed such that the counterfactual action is chosen by the policy. The *minimally complete contrastive explanation* is the explanation that compares only those elements of the explanation tuple that differ. The main drawback of SCM is that it assumes a finite (discrete) state-action space. There is therefore potential for improvement in extending the framework to a continuous state-action space.

Shapley Q-values deep deterministic policy gradient (SQDDPG) was introduced by Wang et al. [42] and is intended for multi-agent settings with continuous action spaces. They extended the centralised critic from multiagent deep deterministic policy gradient (MADDPG [50]) to return a *fair contribution* of the total reward for each agent. This distribution of rewards is estimated by using Shapley values: first, the combined rewards are estimated for all possible subsets of agents (coalitions) using the centralised critic. Then, the contribution for each agent is calculated as the average change in reward that a coalition receives when that agent joins the coalition. The resulting gradient for the objective function is given as:  $\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_s[\nabla_{\theta_i} \mu_{\theta_i}(a|s) \nabla_a Q^{\phi_i}(s, a_i) | a_i = \mu_{\theta_i}(s)]$  where  $Q^{\phi_i}$  is the reward assigned to agent  $i$  as a portion of the total reward and as determined by the Shapley values  $\phi_i$ . Although SQDDPG could help in *characterising* agents based on their contributions to the total reward for a given state-action pair, this method does not provide an explanation for agents’ actions. Therefore, it has potential for improvement.

Riveret et al. [43] introduced probabilistic argumentation (PA); it is a method for classifying RL agents and explaining their actions based on the philosophy of argumentation theory, which is rooted in cognitive science [51]. Using expert knowledge, sets of supporting and attacking arguments are created for each action in a discrete state-action space. Each argument may be comprised of several subarguments. For example, in the game of “*Breakout*” where the agent moves a paddle left or right to bounce an incoming ball such that it destroys bricks, the *supporting* subarguments for the action “right” might include: “*Ball inbound from left*”,  $\neg$  “*Paddle close to right wall*” and “*Ball in Quadrant 3*”<sup>2</sup>, where  $\neg$  represents logical negation. These subarguments are compounded to main arguments, e.g. “*Don’t miss the ball*” or

---

<sup>2</sup>Quadrants are labelled counter-clockwise starting from the north-eastern quadrant

“*Hit the ball through a tunnel*”. While subarguments either attack or support main arguments, the main arguments either support or attack an action. So-called argumentation graphs are then created, which represent all possible combinations of main and subarguments as well as their relationships (attack or support). The subarguments are labelled as “ON” or “OFF” based on the state observation for each time step and the main arguments are labelled as “IN”, “OUT”, or “UNDECIDED” with probabilities learnt in the RL setting:  $(S, A, R)$ , where the states  $S$  are the argumentation graph and the policy to be explained, the actions  $A$  are probabilistic “attitudes” towards arguments, i.e. a probability distribution across the arguments with a sum of 1, and the rewards  $R$  are derived from the argumentation graph and the action taken by the policy for the given state, i.e. supporting arguments for the action will receive higher rewards than attacking arguments. Explanations are obtained from the learnt “attitudes” towards arguments in the argumentation graph, e.g. an agent may prefer not missing the ball in the initial stages of a game, while shifting focus towards tunnelling behaviour in the middle of the game. Surprisingly, this method is not mentioned in recent surveys on explainable RL, such as [2, 3, 4], and shows significant potential.

In methods of the class *hierarchisation and decomposition*, agents’ tasks are decomposed into subtasks, for which simpler and therefore more interpretable policies are learnt. These methods typically require expert knowledge to decompose the problem and therefore pose restrictions on either the types of problem and tasks or the format of the state-action space. However, they inherently simplify the state, action, or rewards spaces of systems, and therefore offer enhanced interpretability. We review four methods in this subclass: Boolean task algebra, reward decomposition, symbolic planning, and hierarchical reinforcement learning (HRL).

Tasse et al. [44] presented a Boolean task algebra that allows decomposition of Boolean tasks into base tasks through standard Boolean operators: conjunction, disjunction, and negation. A task is defined as an objective that an agent must achieve, such as moving to either the left or bottom sides of a grid or collecting all red squares in a map. The authors similarly decompose the value function and show that agents can transfer their knowledge to solving, without any additional training, new tasks such as navigating to the bottom left square in a grid—a Boolean conjunction of the left side and bottom side—, or collecting all blue circles in a map—a Boolean negation of ‘red’ and ‘square’, given that there are only two colours and two shapes of objects on the map. This method imposes certain critical limitations to the MDP, such as: deterministic transition dynamics, reward functions that are constant across tasks except for the terminal states, and perhaps most importantly,

that tasks have a Boolean nature, i.e. the set of possible terminal rewards consists of two values. Decomposition of the value function allows agents to simultaneously learn policies for each of the Boolean states of the reward function. Then, through Boolean algebra, agents may adapt their policies to solve new Boolean combinations of tasks without further training, i.e. through zero-shot transfer learning.

Reward Decomposition, decomposes the reward function into a vector of meaningful reward types. Juozapaitis et al. [46] used such a vectorised reward function in a discrete state-action space where the total reward was the sum of the reward vector. Evaluating the reward vector for each action allows for the classification of actions in terms of reward-based trade-offs. For incomprehensibly large reward vectors, the authors improve interpretability by defining a *minimum sufficient explanation* as the smallest subset of reasons for preferring one action over another, i.e. the smallest subset of values from the reward vector for which the sum is greater than the sums of all equally sized subsets for all competing actions. Similarly, van Seijen et al. [45] decomposed the reward and value functions to exploit domain knowledge for improved training performance. They show that there is an exponential decrease in the size of a problem following reward decomposition and that a decomposition into as little as two or three components can significantly simplify a problem. Consequentially, the simplified reward and value functions allow for simpler explanations and improved interpretability. This method has the potential for being extended to a continuous action space; however, it is doubtful that reward decomposition by itself could satisfy the requirements for explainability. It is more likely that it might serve to characterise agents through reasoning about their motivations for choosing certain actions.

Hierarchical Reinforcement Learning (HRL), sub-divides or decomposes complicated tasks into smaller, simpler tasks. Each type of task is assigned its own agent which, by extension, is also a simpler agent. Each agent is trained to solve a specific type of task while an orchestration agent is trained to choreograph subtasks to accomplish the larger objective. Marzari et al. [47] presented such a solution to train a robotic arm to move objects. They trained three simple DDPG agents to each handle one simple task: moving the arm towards an object in a given location, picking up and placing down an object, and retracting the arm to retrieve the object. They then trained an asynchronous actor-critic (A3C [52]) orchestration agent to activate subtasks in sequence. Although the goal of Marzari et al. [47] was not explainability but improved training performance, the consequence is that each sub-task—and therefore agent—is sufficiently simple and therefore inherently interpretable. However, the agents’ actions are not necessarily inherently explain-

able, especially for the orchestration agent. Further work could therefore be done to provide an explanation for each agent, likely through the use of one of the other methods discussed here.

Lyu et al. [49] proposed a symbolic planning framework, which they call symbolic deep RL (SDRL). Similar to HRL, this framework decomposes the problem into simpler subtasks. It reduces the complexity of high-dimensional state-action spaces by reducing the action space to sets of simple actions. Based on these simple actions, it learns several optimal policies for a sequence of subtasks that achieve specific intrinsic goals. Larger problems are solved by orchestrating the subtasks in a symbolic structure created by human experts; this is where the framework differs from HRL: the symbolic representation takes the form of a tuple  $(I, G, D)$ , where  $I$  is an initial state,  $G$  is an intrinsic goal, and  $D$  is an action description which consists of causal laws created by human experts to associate actions with their effects on the orchestration plan. The set of causal rules within the symbolic representation can lead to more interpretable solutions, since the simple actions are of sufficiently low complexity to be interpretable while the larger orchestrated plan is interpreted through the causal chains designed by the human expert.

## 2.5.2 Inductive Explanations

Since methods in this class rely on generalisations from *observations* they are typically model-agnostic—they are independent of the architecture of the model to be explained. One advantage of this independence is that they have fewer restrictions relating to model architecture and therefore allow, for instance, continuous state-action spaces. However, this comes at the cost of potentially lower fidelity, since the observations may be incomplete samples of the state-action space. While some of these methods rely on simplification through feature saliency or feature extraction, others rely on statistical analyses of observations of the policy. We discuss four methods in this class: state representation learning (SRL), feature saliency methods, introspection, and policy summarisation..

Generally, representation learning aims to learn abstract features that encode information held in an original feature set for purposes such as reducing dimensionality or improving generalisation of a model. Similarly, state representation learning (SRL) learns features from the state space in an RL setting. However, in SRL the encoding of the state space is manipulated by including information from agents' actions, rewards, or other constraints. The goal is to extract useful representations that aid in reasoning about the model's behaviour, and thus facilitate interpretability [34]. Lesort et al. [34] list four approaches to SRL:

**State reconstruction:** States are reconstructed using an autoencoder, resulting in a compressed representation of the state vector which adheres to certain constraints—such as the desired number of dimensions—and can, due to dimensionality reduction, aid in the interpretability of state transitions.

**Forward models:** A forward model predicts the next state from a given state-action pair. Such forward models are often learnt under certain restrictions, such as linearity, which restricts individual state transitions to simple linear dynamics within the learnt state space. Individual state transitions could then be explained in simple terms on the basis of the coefficients of the forward model.

**Inverse models:** An inverse model is trained to predict an action given the state. Such a model not only encodes states, but also learns information needed to reconstruct actions. The encoded state space is thus projected into a more useful representation.

**Prior knowledge:** Using prior knowledge about the environment, such as physics or learnt information about rewards, one may define loss functions related to sets of states. These loss functions are used while training the abstraction model to constrain the embedded space such that states that would otherwise be numerically similar are appropriately separated in the embedded space.

SRL provides the tools and the potential for an explanation, but it does not guarantee explainability and certainly not interpretability. It requires the involvement of the model developer and subject matter experts to extract explanations and form interpretations based on observations in the embedded space.

The use of saliency maps is a well-known method in XAI, specifically to explain convolutional neural networks that deal with image data; they identify areas of images that hold salient information through heat maps overlaid on the images [1]. Greydanus et al. [35] introduced a perturbation-based feature saliency method that measures the change in an agent’s action for a given state following a perturbation of the state’s features. In one example, they used the game “Breakout”, where the agent has to destroy bricks by bouncing a ball off a paddle. They used A3C [52] to train their agent in a discrete action space. After training, they measured the change in the agent’s action when perturbing the pixels of the input frame. They did this for both the actor and the critic and, through saliency maps, showed that while the actor placed most weight on the positions of the paddle and ball, the critic focused on the location of a potential tunnel in the layer of bricks. This disclosed the strategies of both actor and critic: while the actor had learnt the

importance of not missing the ball with the paddle, the critic had learnt the value of placing the ball through a tunnel and scoring compounded points. This work has the potential to be extended to continuous action spaces, as input perturbation while measuring the agent’s response is rather generic. However, large state spaces might effect the interpretability of feature saliency methods, in which case they might have to be combined with other methods, such as dimensionality reduction through SRL, or game theoretic perturbation through Shapley values. Furthermore, this method lacks the *causal chain* offered by SCM and can only provide feature saliency estimates for single time steps.

Introspection is based on statistical analyses of the history of the agent’s experience in the environment, either during or after training [36]. Specifically, the following data are analysed:

- $n_s, n_{s,a}, n_{s,a,s'}$  the number of times a given state ( $s$ ), state-action pair ( $s, a$ ), and transition ( $s, a, s'$ ) had occurred
- $P(s'|s, a)$  the probability of transitioning to  $s'$  when performing action  $a$  in state  $s$
- $\hat{R}(s, a)$  the estimated reward for performing action  $a$  in state  $s$
- $Q_\pi(s, a)$  the expected value of a state-action pair given the current policy  $\pi$
- $\widehat{\Delta Q}(s, a) = r(s, a) + \gamma \max_{a' \in \mathcal{A}} Q(s', a') - Q(s, a)$  the estimated prediction error of the value of a given state-action pair. During training, the agent can keep track of the estimated prediction error for all states and actions, given a finite (discrete) state-action space.

The analysis of the data aims to extract so-called *interesting* elements, such as exceptionally certain/uncertain transitions, mean rewards and reward outliers, state-space coverage (how much of the state space has been visited) and evenness (uniformity of the distribution of visits to states), frequent / infrequent visits to states, strongly / weakly associated feature sets (features in the state with high / low variability), state-action value outliers, mean prediction error, mean prediction outliers, etc. One of the benefits of this framework is that it provides an online monitoring mechanism where an agent can monitor these *interesting* elements and, e.g. proactively request human assistance in situations where actions are uncertain.

Amir et al. [37] proposed a *conceptual* framework in which an agent’s policy might be summarised and presented to an audience as an explanation. They list three elements that constitute such a summary:

**Intelligent state extraction:** Given an agent’s policy, a subset of state-action pairs is selected to be included in the explanation. These state-action pairs are chosen based on criteria determined in one of three approaches:

- *States of interest* in which the most salient states are chosen based on either the value function, the degree of coverage of the entire state space or the likelihood that a state is encountered.
- *Policy reconstruction accuracy* which requires that a human be able to reconstruct the policy, given only this subset of state-action pairs.
- *Peer designed agents* in which rule-based agents are created by humans. These rules are then used to determine which states were regarded as important by a human subject matter expert.

**World-state representation:** The authors state that encoding the state representation such that salient information of the state is effectively conveyed to a human audience is not trivial. They propose enlisting the help of domain experts to manually reduce the dimensionality of the state. This is a potential oversight as some of the other methods we mention may aid in this endeavour, e.g. SRL, simplification through decomposition, or Shapley values. This is a potential area of improvement in the proposed conceptual framework.

**Strategy summary interface:** is a proposed interactive interface in which a user can review and explore agent policies. The authors list three considerations to achieve this objective:

- *Summary presentation* which considers the appropriate form in which a policy summary is presented, i.e. visualisation through graphs, heat maps, etc., or textual presentation using natural language.
- *User guided exploration of policies* which proposes a collaborative interface where users might effect the state representation or even the policy itself to explore potential counterfactual actions.
- *Understanding users’ extrapolation of summaries* which suggests testing the assumptions made regarding the depth of knowledge of the users, the biases, and the reasoning when interpreting the explanations.

The authors propose no methods towards realisation of these objectives but simply list them as considerations. The framework nevertheless has potential for further development.

### 2.5.3 Summary

The recent advancements in RL have increased its complexity, which has led to challenges related to explainability, interpretability and therefore understanding [2]. To mitigate this, there have been several attempts at explaining the actions of agents and interpreting their behaviour resulting in a wide suite of explainability approaches. We propose a taxonomy of the most recent methods in explainable RL; each of these methods has inherent advantages and disadvantages, which we list in Table 2.1. Our proposed taxonomy reveals commonalities in these inherent properties, e.g. methods in the class *explanation and policy integration* have the potential for high fidelity but can be difficult to interpret for non-experts, while methods in the class *hierarchisation or decomposition* are typically highly interpretable. It is our hypothesis that, through a unification of selected methods, one may mitigate their disadvantages while retaining their desirable properties. Such methods should be selected from complementing classes in our taxonomy. Two methods that show this potential for improvement are structural causal modelling and probabilistic argumentation; they both produce explanations with high fidelity but demand finite state-action spaces which could be obtained through simplification using, e.g. SRL.

## 2.6 Potential Research Opportunities

Reward functions represent an often complex rendition of expert domain knowledge and creating a suitable reward function can therefore be nontrivial. This is exacerbated by practices such as reward shaping that add elaborate complications to the reward function to encourage agents to learn the intentions of an expert. It also amplifies issues such as reward hacking in which agents learn the literal reward function instead of generalising the expert’s intention. Some approaches, such as constrained RL, can further complicate the reward function, while others, such as preference-based RL, replace the reward function with similarly complex action-preference relationships. In contrast, intrinsic motivation, and specifically policy regularisation, expedites the learning process by guiding the policy to select, e.g. a uniform action distribution, thus encouraging exploration. This eliminates the need for over-complicated utility functions, while ensuring agents act as intended. It is not inconceivable that regularisation could also be used to instil domain-specific, desirable behaviours. We develop such a method, affinity-based RL, which we conceptualise in Chapter 3.

The uncertainties that exist in environments are a major catalyst for the need for



Table 2.1: An overview of recent approaches in explainable reinforcement learning.

Classification	Method	References	Advantages*	Disadvantages*
Deductive: Explanation & Policy Integration	Symbolic AI	[38, 39, 40]	(ii), (iii)	(b), (c)
	Structural causal modelling (SCM)	[41]	(i), (iv), (v)	(a), (d)
	Shapley Q-values DDPG (SQDDPG)	[42]	(i), (iii)	(a), (c)
	Probabilistic argumentation (PA)	[43]	(i), (vii)	(a), (b), (d), (f)
Deductive: Hierarchisation & Decomposition	Boolean task algebra	[44]	-	(a), (b), (c), (d)
	Reward decomposition	[45, 46, 47]	(ii)	(b), (c), (d)
	Hierarchical reinforcement learning (HRL)	[48, 47]	(ii), (iii)	(c)
	Symbolic planning (SDRL)	[49]	(ii), (iii)	(b)
Inductive Explanations	State representation learning (SRL)	[34]	(ii), (iii)	(b), (c)
	Feature saliency	[35]	(iii)	-
	Introspection	[36]	(iii), (vi)	(c)
	Policy Summarization	[37]	(iii), (vii)	(b), (f)

\*Advantages and disadvantages are enumerated in Table 2.2

Table 2.2: Lists of advantages and disadvantages for methods in Table 2.1.

Advantages	Disadvantages
(i) Explanations are conjugate to policies and have potential for very high fidelity	(a) Explanations can be difficult to interpret for non-experts
(ii) Simplifies the state / action / reward space which improves interpretability	(b) Requires expert domain knowledge
(iii) Allows for continuous state-action spaces	(c) Facilitates, but does not guarantee explainability
(iv) Uses causal chains, i.e. future states are included in the explanation	(d) Assumes finite, discrete state-action spaces
(v) Allows counterfactual explanations	(e) Assumes Boolean decomposable tasks
(vi) An agent can report on the uncertainty of their actions in real time	(f) Presented as a concept but has not been proven in practice
(vii) A potential framework for unification of several other methods	

explainability in systems trained with RL. To date, most, if not all, of the work on explainable RL has focused on post-hoc explanations and interpretations of trained models. The complexity of AI models poses serious challenges to the fidelity and validation of the extracted explanations; it also further amplifies any uncertainties that may exist in an environment. Our proposed affinity-based RL principle intrinsically incorporates desirable properties during training and thus makes a contribution towards interpretability that does not depend on any particular architecture or learning method. In the long run, explainability and interpretability—and in fact verification—of future RL systems are imperative if they are to be deployed in critical applications that demand high degrees of trustworthiness.

While HRL is a proven approach for *sequential* composition of strategies, we

rather propose a fuzzy superposition of elemental strategies. The Arnold–Kolmogorov representation theorem states that any continuous multivariate function can be represented by the superposition of univariate functions [53], and Hilbert’s thirteenth problem reconstructs multivariate functions from bivariate functions [54]. Using affinity-based RL, we design prototypical policies in which agents act according to defined action associations. We then combine these prototypical strategies to form bespoke superpositions that solve problems in unique and tailored ways. These strategies are naturally interpretable according to the amalgam of their elemental prototypical behaviours.



# Chapter 3

## Principles of Affinity-Based Reinforcement Learning

### 3.1 Conceptual Formulation

Affinity-based reinforcement learning (ab-RL) is a new paradigm that learns prototypical strategies that are interpretable according to defined action affinities. It is a parsimonious and transparent procedure that addresses unheeded issues such as the complexity of preferences in preference-based RL and the opacity of constrained RL that also neglects *desirable* strategies. In this chapter, we conceptualise and construe the generic ab-RL framework through a theoretical formulation and an illustrative example.

The ab-RL framework defines the following elements: (1) *traits* are the elemental attributes that describe the prototypical qualities of an individual or entity, (2) *associations* are the relationships between traits and actions, and (3) *affinities* are the preferences of prototypical agents toward certain actions, specified as probabilistic prior action distributions  $\pi_0^i$ ,  $i \in [1, N]$  where  $N$  is the number of prototypical agents. The learnt policies  $\pi^i$  of these prototypical agents are finally combined to form fuzzy superpositions that may vary in time, i.e.  $\pi^* = \omega_1(t)\pi^1 + \omega_2(t)\pi^2 + \dots + \omega_N(t)\pi^N$  where  $\omega_i(t)$  are weighting terms that are functions of time. We illustrate the generic ab-RL framework in Figure 3.1. To the best of our knowledge, this type of fuzzy composition of prototypical strategies has never been achieved. It follows the principles of the Arnold-Kolmogorov<sup>1</sup> representation theorem and Hilbert’s thirteenth problem<sup>2</sup> [53, 54]; we are able to reason about this superposition of elemental functions through understanding the sum of its parts.

---

<sup>1</sup>The Arnold–Kolmogorov representation theorem states that every multivariate continuous function can be represented as a superposition of univariate functions.

<sup>2</sup>Hilbert’s thirteenth problem involves the reconstruction of a seventh order function from bivariate functions.

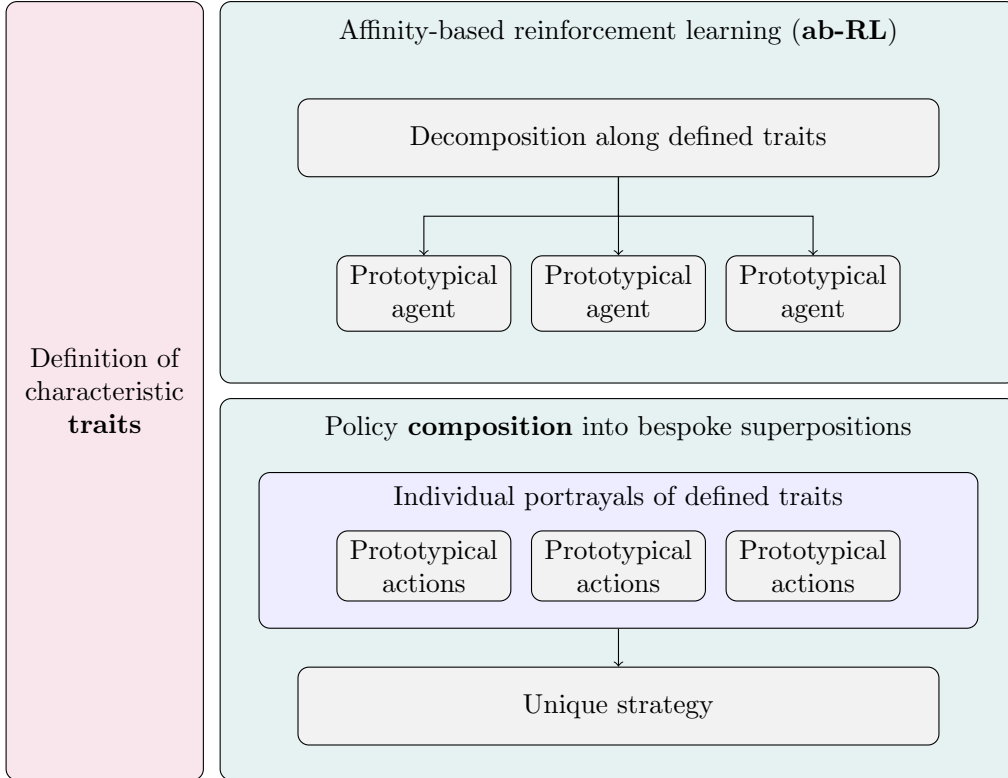


Figure 3.1: Illustration of the affinity-based reinforcement learning framework. Task decomposition is done according to prototypical associations between characteristic traits and agent actions. These associations lead to action affinities, i.e. preferred action distributions, that are also the interpretations of agents’ behaviour. These prototypical strategies are finally composed to create unique strategies that are interpretable as weighted averages of the prototypical strategies.

## 3.2 An Illustrative Example of Affinity-Based RL

We demonstrate the utility of ab-RL in a simple example in Figure 3.2. In this example, an agent navigates a grid in search of a randomly generated destination. The set of traits that define agents’ characteristics is  $\mathcal{T} = \{safety, efficiency\}$ . Consider the premise that right turns are safer than left turns and that the shortest route is naturally more efficient [55]; the trait *safety* might therefore highly associate with right turns, while the trait *efficiency* might highly associate with going straight. The agent shown in Figure 3.2 is the prototypical agent representing *safety*; its affinity is defined as  $\pi_0^{safety} = [0, 0.4, 0.6]$ , where the values represent the probabilities of turning left, going straight, and turning right, respectively. It is clear from Figure 3.2a that this agent prefers right turns, even if such actions result in less efficient routes. Another prototypical agent, which is not shown in Figure 3.2, could represent *efficiency* with a high affinity for going straight, e.g.  $\pi_0^{efficiency} = [0.1, 0.8, 0.1]$ . A useful superposition of their prototypical policies  $\pi^* = \sum_{i \in \mathcal{T}} \omega_i(t) \pi^i$  might, for ex-

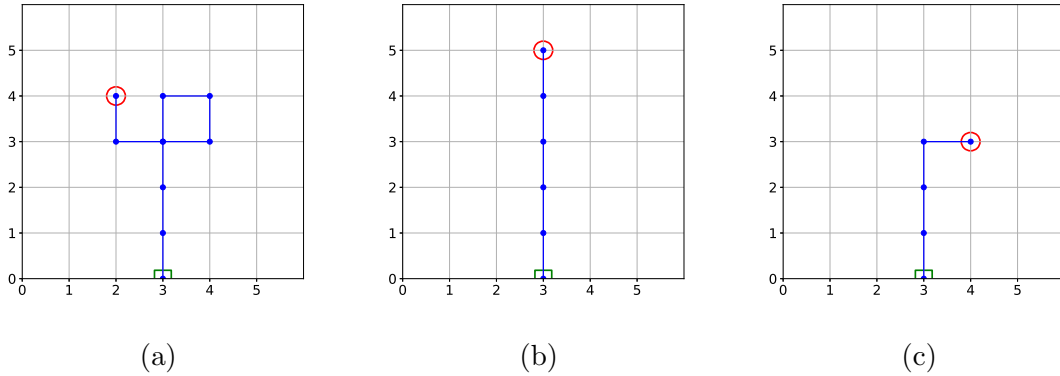


Figure 3.2: An illustrative example of the prototypical behaviour of an agent with an affinity for right turns in a grid navigation problem. The initial location is marked with a green square at coordinates (3,0), and the destinations are marked by red circles. In (a) the agent makes a series of right turns to reach its destination on the left, even though the shortest path would have involved a single left turn, in (b) it needs not turn, and in (c) it follows the shortest path involving a single right turn. Figure taken from Maree and Omlin [56].

ample, dictate that an agent acts safely during rush hour and efficiently otherwise. Another example is that of virtuous agents, where an agent learns to act bravely or honourably depending on the situation [17] (refer to Appendix I).

### 3.3 Agent Prototyping with Affinity-Based RL

In ab-RL an agent learns a prototypical behaviour through regularisation of its objective function, given a set of action affinities. Regularisation encourages an agent to behave according to the prior, which makes the policy inherently interpretable. Although policy gradient methods generally do not guarantee a globally optimal solution, they do guarantee a locally optimal solution [19]. Regularisation defines that region in the state-action space where this locally-optimum solution shall be found; while entropy regularisation encourages *wide exploration* by specifying a uniform action distribution, ab-RL encourages *specific exploration* by defining a specific action distribution. The convergence guarantees of ab-RL follow from the guarantees provided by policy regularisation in general [57]. In ab-RL, we penalise the objective function whenever the action distribution deviates from a desired global prior:

$$\begin{aligned}
 J(\theta) &= \mathbb{E}_{s,a \sim \mathcal{D}} [R(s, a)] - \lambda L & (3.1) \\
 L &= \frac{1}{M} \sum_{j=0}^M \left[ \mathbb{E}_{a \sim \pi_\theta} [a_j] - (a_j | \pi_0(a)) \right]^2
 \end{aligned}$$

where  $\mathcal{D}$  is the replay buffer and  $L$  is the regularisation term, which is the mean square difference between the current action distribution and the prior action distribution  $\pi_0$ , across  $M$  number of actions [56] (refer to Appendix E). We detail the policy regularisation algorithm in Algorithm 1.

---

**Algorithm 1** Policy regularisation algorithm, taken from Maree and Omlin [58].

---

```

Initialize the actor  $\mu_{\theta_\mu}$  with random parameters  $\theta_\mu$ 
Initialize the critic  $Q_{\theta_Q}$  with random parameters  $\theta_Q$ 
Initialize the target actor  $\mu'_{\theta_{\mu'}}$  with parameters  $\theta_{\mu'} \leftarrow \theta_\mu$ 
Initialize the target critic  $Q'_{\theta_{Q'}}$  with parameters  $\theta_{Q'} \leftarrow \theta_Q$ 
Set the prior  $\pi_0$  and the number of actions  $M_i \leftarrow |\pi_0|$ 
Set regularisation weight hyperparameter  $\lambda$ 
Set target update rate hyperparameter  $\tau$ 
Initialize the replay buffer  $\mathcal{D}$ 
for  $e = 1$ , episodes do
  Initialise a random exploration function  $F(e) \sim N(0, \sigma_e)$ 
  Reset the environment and get the first state observation  $s_1$ 
   $t \leftarrow 1$ ,  $Done \leftarrow False$ 
  while not Done do
    Select the action and add exploration randomness  $a_t \leftarrow \mu_{\theta_\mu}(s_t) + F(e)$  ▷ Gather experience
    Retrieve the environmental response: reward  $r_t$  and observation  $s'_t$ 
    Store the transition tuple  $\mathcal{T} = (s_t, a_t, r_t, s'_t)$  to replay buffer:  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{T}$ 
     $t \leftarrow t + 1$ 
     $s_t \leftarrow s'_t$ 
    if (end of episode) then
       $Done \leftarrow True$ 
    end if
  end while
  Sample a random batch from the replay buffer  $\mathcal{B} \subset \mathcal{D}$  ▷ Learn using experience replay
   $\hat{Q} \leftarrow r_{\mathcal{B}} + \gamma Q'_{\theta_{Q'}}(s_{\mathcal{B}}, \mu'_{\theta_{\mu'}})$ 
  Update critic parameters  $\theta_Q$  by minimising the loss:


$$\mathcal{L}(\theta_Q) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} (Q_{\theta_Q} - \hat{Q})^2$$


  Update the actor parameters  $\theta_\mu$  by minimising the loss: ▷ From Equation 3.1


$$\mathcal{L}(\theta_\mu) = -\bar{Q} + \lambda \frac{1}{M} \sum_{j=1}^M [\bar{\mu}_j - (a_j | \pi_0)]^2$$


  Update the target parameters:


$$\theta_{\mu'} \leftarrow \tau \theta_\mu + (1 - \tau) \theta_{\mu'}$$


$$\theta_{Q'} \leftarrow \tau \theta_Q + (1 - \tau) \theta_{Q'}$$


```

---

**end for**

---

Although affinity-based agents are inherently interpretable, they lack a symbolic explanation of their policies. One way to extract these explanations is through global surrogate modelling using Markov models. A hidden Markov model (HMM) learns the transition probabilities of an unobservable Markov process  $X$  by observing a Markov process  $Y$  with the property that its current state  $Y_t \in Y$  is solely dependent on the state  $X_t \in X$ , which is solely dependent on the previous state  $X_{t-1} \in X$  (the Markovian property) [59]. These transition probabilities are represented by a Markov matrix  $F_{ij} = P(X_{n+1} = j \mid X_n = i)$ , where the values in the rows in  $F$  add up to one. There exists a similar Markov matrix  $E$  that describes the state

observations, or emission probabilities,  $E_{ij} = P(Y_t = j \mid X_t = i)$ . We illustrate this process in Figure 3.3. If given only a series of states  $\{Y_t\}_{t=0}^T$ , the transition matrix  $F$  and the emission matrix  $E$  can be estimated using the Baum-Welch algorithm—a special case of the expectation-maximisation algorithm [60]. If, in addition, the states  $\{X_t\}_{t=0}^T$  are also known, the matrices  $F$  and  $E$  can be calculated directly. These matrices are the symbolic explanations of each prototypical agent’s policy.

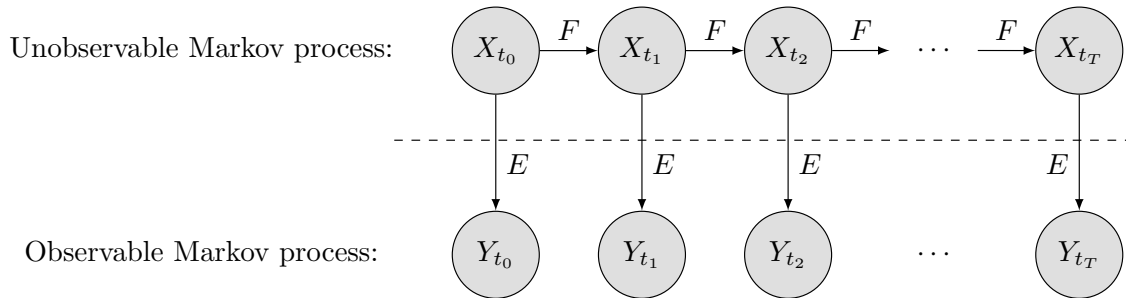


Figure 3.3: A diagram representing a hidden Markov model. An unobservable Markov process  $X$  and observable Markov process  $Y$  are described by a transition probability matrix  $F$ , and emission probability matrix  $E$ . Taken from Maree and Omlin [61].

### 3.4 Composition of Traits and Policies

Task decomposition along traits is a domain-specific endeavour. However, it is natural for individual portrayals of defining traits to fluctuate over time. In our grid navigation example, a rational agent might show safe tendencies at times, and efficient ones otherwise. In a different example, our activity on social media can be used to predict our personality traits [62]. Such activity data are naturally high-dimensional, sparse, and time-dependent. The aim is to extract useful, low-dimensional representations of behavioural indicators that underlie and predict characteristic traits. Recurrent neural networks (RNNs) are useful to extract temporal features that represent individual portrayals of characteristic traits over time. RNNs generally perform well in low-dimensional representations and, therefore, can also help reduce the dimensionality of sophisticated behaviours [63]. The features extracted from such RNNs naturally form useful inputs for composing superpositions of prototypical policies; they result in the time-dependent weights  $\omega_i(t)$  that combine prototypical policies. These weights can be predicted in a new RNN that takes as input the extracted temporal features.

We detail our methodology of using ab-RL to solve the problem of personal financial advice in Chapter 4 and show the results in Chapter 5.





# Chapter 4

## Empirical Methodology

### 4.1 Empirical Justification for Affinity-Based RL

In [24], we have trained a hierarchical orchestration of prototypical agents that does not make use of ab-RL (refer to Appendix D). Instead, these agents learnt to manage a portfolio of stocks through unique reward functions. While a profit-aware agent maximised daily returns, a risk-aware agent maximised the Sharpe ratio<sup>1</sup>. An orchestration agent then learnt a linear composition of these two prototypical agents that optimises the mean ESG<sup>2</sup> score of the portfolio. These prototypical agents’ policies are—perhaps crudely—interpretable, i.e. they choose their actions according to their specific reward functions that make them either profit-aware, risk-aware, or sustainability-aware. However, policy gradient methods do not guarantee convergence to a global optimum [19]. Instead, they merely guarantee a local optimum, and we observed a unique local optimum for each unique set of initial parameters. This was due to flat policy gradients preventing gradient descent from finding a suitable optimum, and we reverted to genetic algorithms to find optimum policies. Flat policy gradients seem to be a general issue for portfolio optimisation problems [24]. The reason could be that the agents’ actions—buy and sell—do not directly affect the state—changes in stock prices—and, therefore, the gradient of the objective function with respect to model parameters  $J(\theta) = \mathbb{E}_{s,a \sim \mathcal{D}} [R(s, a)]$  diminishes for large batches of training data  $\mathcal{D}$ . Adding an action-dependent regularisation term  $L(\pi, \pi_0)$  to the objective function,  $J(\theta) = \mathbb{E}_{s,a \sim \mathcal{D}} [R(s, a)] - \lambda L(\pi, \pi_0)$ , improves convergence by improving causality between the action distribution  $\pi$  and the value of the objective function  $J(\theta)$ . Affinity-based RL thus addresses concerns relating to

---

<sup>1</sup>The Sharpe ratio is a popular measure of risk in a portfolio, with higher values indicating a lower risk to reward ratio.

<sup>2</sup>The ESG—environmental, social, and governance—score is a popular sustainability measure of a company.

vanishing policy gradients and guarantees convergence to a unique local optimum that depends on the prior action distribution [15].

## 4.2 Application of Affinity-Based RL in Finance

### 4.2.1 Application Overview

We demonstrate the utility of ab-RL in an application in finance. Using the classified financial transactions of ca. 26,000 real customers over a six-year period, we created an interpretable AI for personal investment management. This system generates personal investment strategies based on individuals' spending behaviour. Financial transactions are a predictor of financial personality, which affects how we invest [64, 65]. We used ab-RL to imbue agents' prototypical policies with the characteristic behaviours considered desirable to the different customer personality traits. These traits are the dimensions of a popular personality model (openness, conscientiousness, agreeableness, extraversion, and neuroticism), which we discuss in Section 4.2.2. We extracted features from a RNN that predicts these traits using customers' financial transactions as input, and used these extracted features as the input to another RNN that composes superpositions of our prototypical agents. We illustrate the application of ab-RL to personal investment advice in Figure 4.1, and in the following subsections detail our methodology for this application.

### 4.2.2 Feature Extraction with RNN

Customer segmentation is a non-trivial task that requires the consideration of time-variant behavioural patterns [66]. To this end, we extracted the temporal features that describe customers' spending personalities from the node activations of a three-node RNN, hereafter referred to as the state space of the RNN. Personality is commonly modelled using five *traits* that capture individual differences across cultures, locations, languages, etc. These five traits constitute the Big Five personality model: (1) openness, being open to new experience; (2) conscientiousness, the tendency to be organised, have self discipline, and aim for achievement; (3) extraversion, seeking stimulus in the company of others; (4) agreeableness, the propensity for cooperation and compassion, and (5) neuroticism, the tendency to more easily experience unpleasant emotions [9]. These are the traits that define the five prototypical agents in our application. Our RNN predicts these five traits from the classified financial transactions of customers; the inputs are annually-aggregated transaction values across  $N = 97$  transaction classes over  $T = 6$  years. Each value in the input vec-

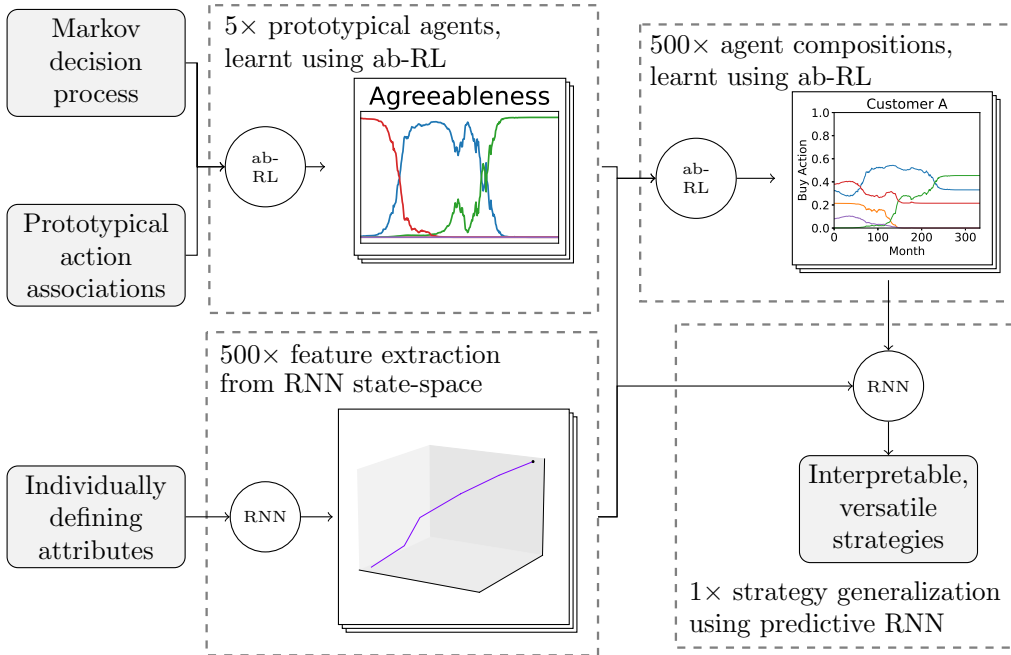


Figure 4.1: A flow diagram illustrating our methodology of decomposition and composition of prototypical strategies. From a defined MDP and decomposed prototypical action associations, we create five affinity-based RL agents that each associate with one defining attribute. We combine these policies to form superpositions, resulting in individual strategies that match the time-variant, continuous segmentation of 500 individuals; these superpositions are inherently interpretable and continuously variable according to individuals’ changing characteristics in time. We use a RNN to predict the weights that combine the five prototypical agents for the 500 individuals, thus generalising agent composition. Figure adapted from Maree and Omlin [61].

tor  $x_i = \{x_{i,t}, t \in [1, T]\}$ ,  $i \in [1, N]$ ,  $\sum_{i=0}^N x_{i,t} = 1$  represents the fraction of the total annual expenditure in the category  $x_i$  for year  $t$ . We illustrate this model architecture in Figure 4.2.

The extracted features represent the encoded spending behaviours that classify customers according to the five personality traits. We defined the dominant personality trait as that trait in which a customer has the highest degree of membership, and we observed a hierarchical clustering of the feature trajectories along successive levels of dominance of the personality traits [67] (refer to Appendix B). To interpret this behaviour, we refer to the theory of dynamical systems; it explains the evolution of the state of a system, where state dimensions represent the system variables, and the motion of points in the state space reflects changes in the values of these variables. An example is the swing of a pendulum of which the state is described by its angle and angular velocity. An attracting set, or attractor, is a point, or set of points, in the state space toward which a system will evolve from many different initial coordinates. A pendulum, for example, will eventually evolve to the neutral position, irrespective of its initial state. Attractors typically convey useful

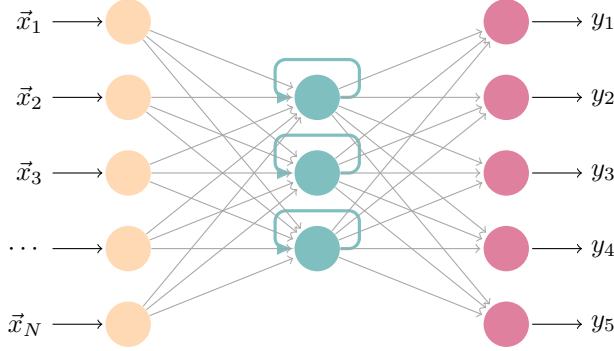


Figure 4.2: An illustration of the recurrent neural network architecture that we used for extracting temporal features representing customer spending behaviour. The input vectors  $x_i$ ,  $i \in [1, N]$  represent the fractions of total annual spending in each category across six years of observed data. The outputs  $y_j$ ,  $j \in [1, 5]$  are the degrees of membership in each of the five personality traits. The extracted features are the activations of the three hidden nodes.

information about their dynamical systems [68]. We considered the state space of our RNN and observed that the features formed trajectories that changed direction only when a customer changed their spending behaviour. We located the attractors that govern the dynamics of this state space and labelled them according to each of the five personality traits [69] (refer to Appendix C). The interpretation of our model is that each personality trait has a corresponding attractive set that acts on the feature trajectories according to customers’ degree of membership in each of the personality traits; the higher the degree of membership in a given trait, the larger the attraction to the corresponding attractive set. This interpretation is important for the remainder of our work and provides insight into trajectory behaviour in relation to personality profiles.

### 4.2.3 Affinity-Based RL for Personal Investment Advice

Having identified the five defining personality traits of financial customers, we applied ab-RL to train five prototypical agents to invest in a set of asset classes: stocks, property, savings accounts, mortgage curtailment, and luxury items. While the investment classes stocks, property, and savings accounts are self-explanatory, we define mortgage curtailment as payments that reduce the principal debt of the loan, and luxury items as items defined in, e.g. the Knight-Frank luxury investment index [70]. We employed a panel of domain experts to determine the *associations* between the personality traits and these asset classes; they correlated the inherent asset class properties to the preferences of the personality traits, as shown in Table 4.1. From this table, we see that conscientiousness, for example, is highly associated with reduced risk, while openness is highly associated with perceived

novelty and high liquidity of assets. The same panel of experts then ranked the asset classes according to their expected long-term performance in the same set of properties, which we show in Table 4.2. We then calculated a set of coefficients  $C$

Table 4.1: Matrix  $A$  containing a set of asset class properties and their associations with the five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. The values are in the set  $\{n \in \mathbb{Z} \mid -2 \leq n \leq 2\}$  and indicate a strong negative, slightly negative, neutral, slightly positive and strong positive association, respectively. Taken from Maree and Omlin [71].

Asset class property	Open.	Cons.	Extra.	Agree.	Neur.
High returns	1	1	2	1	1
High liquidity	2	-1	2	1	2
Low capital prerequisite	0	-1	1	1	1
Low risk	-1	2	-1	1	2
High novelty	2	0	2	0	-1

Table 4.2: Matrix  $B$  containing ratings for the asset classes with regard to a set of properties. The values are in the range  $[0, 1]$  and higher values represent higher performance in each of the asset class properties. Taken from Maree and Omlin [71].

Asset class property	Savings	Property	Stocks	Luxury	Mortgage
High returns	0.25	0.67	1.00	0.05	0.50
High liquidity	1.00	0.25	0.80	0.10	0.05
Low capital prerequisite	0.80	0.25	1.00	0.50	1.00
Low risk	1.00	0.32	0.10	0.05	1.00
High novelty	0.10	0.25	0.75	1.00	0.10

that directly associate asset classes with personality traits using matrix multiplication:  $C = (A^T \cdot B^T)^T$ . These coefficients, scaled to the range  $[-1, 1]$  and shown in Table 4.3, quantify personality-based *affinities* towards different asset classes. These affinities are the regularisation priors  $\pi_0^i$ ,  $i \in [1, 5]$  from which the agents learnt their locally optimal policies, i.e. the prototypical investment strategies.

Table 4.3: Coefficients, in the range  $[-1, 1]$ , associating asset classes to prototypical personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. Higher values indicate where personality traits might have higher affinities towards asset classes. Taken from Maree and Omlin [71].

Asset type	Open.	Cons.	Extra.	Agree.	Neuro.
Savings account	-0.11	0.08	-0.15	0.51	0.68
Property funds	-0.15	0.32	-0.22	-0.36	-0.24
Stock portfolio	0.82	-0.61	0.95	0.42	0.12
Luxury expenses	0.16	-0.51	-0.07	-0.80	-0.81
Mortgage repayments	-0.72	0.72	-0.52	0.23	0.25

We finally extracted symbolic explanations for these interpretable prototypical strategies using Markov models in [61] (refer to Appendix H).

#### 4.2.4 Agent Composition using RNN

We used ab-RL to learn the optimal combinations of the prototypical agents for 500 different customers, i.e.  $\pi^{*j}$ ,  $j \in [0, 500]$ . Here, the individual priors  $\pi_0^j = \mathcal{P}_k$  were the customers’ fuzzy memberships  $\mathcal{P}_k$  in each of the five personality traits  $k \in K = \{\textit{openness}, \textit{conscientiousness}, \textit{extraversion}, \textit{agreeableness}, \textit{neuroticism}\}$ ; we regularised the compositions to closely reflect the customers’ personality profiles  $\mathcal{P}$ . This resulted in 500 ab-RL agents whose actions are the weights  $\omega_k$  that govern superpositions of the prototypical policies. Due to our customer data being limited to the last six years’ transaction history, we formulated our MDP such that the state was the static spending behaviour of each of the 500 customers, i.e. the feature trajectory that represented the last six years’ spending personality for each customer. This resulted in consistent weights across all time steps throughout the episode, i.e. the final policies were the consistent weighted averages of the prototypical policies  $\pi^{*j} = \sum_{k \in K} \omega_k^j \pi^k$ . Intuitively, these agents learnt how to combine prototypical strategies for customers who never changed their spending behaviour [58] (refer to Appendix F).

To obtain time-variant strategies, we trained a RNN to predict these weights  $\omega_k^j$  from customers’ spending behaviour; this RNN had three recurrent nodes and its inputs were the same extracted feature trajectories used in the aforementioned MDP. This is a generalisation of strategy composition that predicts the time-variant compositions  $\omega_k^j(t)$  that adapt as spending behaviours fluctuate in time; when a customer changes their spending habits, the composed investment strategy changes to accommodate their new interests and preferences. Therefore, for each time step  $t$ , the RNN uses a moving window of the last six years’ data to predict the weights that combine the prototypical agents  $\omega_k(t)$ ,  $t \in [1, T]$ , where  $T$  is the time horizon [71]. In general, the weights change slowly, because consistent spending patterns change slowly compared to the sample rate of the investment agent (refer to Appendix G).

# Chapter 5

## Application in Financial Advising

### 5.1 Explainable Transaction Classification

We make a distinction between explainability and interpretability [72] (see Appendix A). Most authors use these terms interchangeably and distinguish between different levels of explainability through the concept of a target audience. We rather define explainability as a *symbolic representation* of a model’s predictions and interpretability as the *tools needed for humans to reason* about a model’s behaviour. These definitions eliminate the requirement of a target audience, as “symbolic explanations” are by definition concrete formulations while “the tools needed for human reasoning” implies different tools for different cognitive backgrounds. We illustrate the distinction between these concepts in a simple application, where we use existing XAI practises to extract explanations and interpretations from a transaction classification model that is currently in production at a major Norwegian bank.

Our spending patterns are a predictor of our personalities [9]. These spending patterns become evident through financial transactions classified into categories such as groceries, travel, transport, etc. Any AI that uses these classifications cannot claim to be transparent if the classification model is opaque. We clustered the classified transactions according to the most salient feature—transaction text—and labelled these clusters with the known transaction categories. We extracted common keywords that are the *interpretations* of the classification, e.g. a transaction containing the text “supermarket” is classified as “groceries”; nonhomogeneous clusters also existed, e.g. the keyword “Shell” can result in classifications of either “fuel” or “kiosk”. We used shallow decision trees to distinguish between such nonhomogeneous clusters. These decision trees are the symbolic *explanations* of the model.



## 5.2 Customer Micro-Segmentation

Financial transactions are manifestations of our digital footprints that reveal some of our personality traits. Tovanich et al. [66] have shown that financial transactions over time offer superior classification compared to their non-temporal components. We exploit this increased accuracy by predicting customers' personalities with a RNN from customers' classified financial transactions over time [67] (refer to Appendix B). The state space of this RNN reveals feature trajectories, which form a hierarchy of subclusters along successively less dominant personality traits. This hierarchy is important, as it suggests a micro-segmentation of customer behaviour; the locations of the trajectories within the clusters are significant. Figure 5.1 shows the 3-dimensional state space of our RNN with a subset of trajectories, each representing a customer. We labelled the trajectories according to customers' dominant personality traits and observed clustering of these trajectories (Figure 5.1a). Figure 5.1b shows two trajectories for the same customer with a consistent spending behaviour during the observation period (2014-2019); it shows two trajectories for two different time periods, one for six years (2014-2019) and one for one year (2019). The customer's consistent spending behaviour—and evinced personality profile—is reflected in the similar paths of these two trajectories. In contrast, Figure 5.1c shows two such trajectories for a different customer, who changed their spending behaviour in 2019 such that their dominant personality trait changed from neuroticism to conscientiousness; the long-term trajectory reflects an abrupt change in spending behaviour and, in 2019, follows a direction similar to the trajectory that corresponds to 2019.

The dynamics of these trajectories are interpreted through the theory of dynamical systems; trajectories are attracted to sets of points in the state space that are associated with different spending behaviours and personalities. We located these attractors and labelled them according to their associated personality traits Maree and Omlin [69] (refer to Appendix C). Figure 5.2 shows how trajectories asymptotically converge to these labelled attractors. Three point attractors correspond to the personality trait conscientiousness, and trajectories converge to each of these depending on initial conditions. There are three line attractors that each correspond to agreeableness, extraversion, and neuroticism, respectively. Finally, a single point attractor corresponds to the trait openness. It should be noted that no distinction is made between line and point attractors, nor is there a significance behind one personality trait corresponding to three distinct point attractors [68]. Each basin of attraction holds a cluster of trajectories that each form a hierarchy

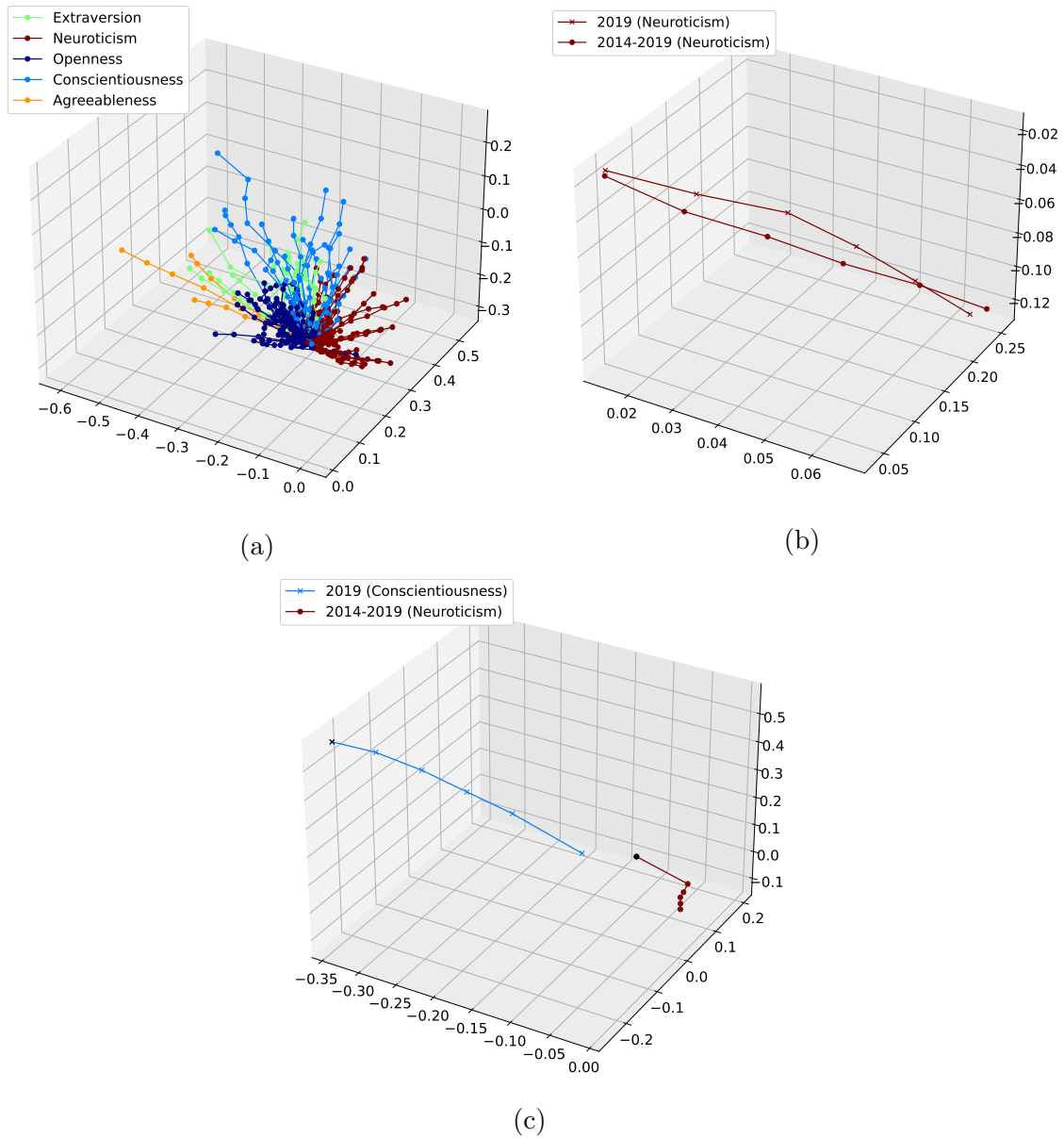


Figure 5.1: Two trajectories in the 3-dimensional state space of a recurrent neural network trained to predict personality from aggregated transactions of a single customer. While (a) shows the clustering of the trajectories of many customers according to their most dominant personality traits, (b) shows two trajectories for the same customer identically classified for two different time periods: one year vs. six years, and (c) shows two trajectories for another customer who changed their spending behaviour in the sixth year such that the trajectory converges to a different attractor (conscientiousness) than the first five years (neuroticism). Taken from Maree and Omlin [69].

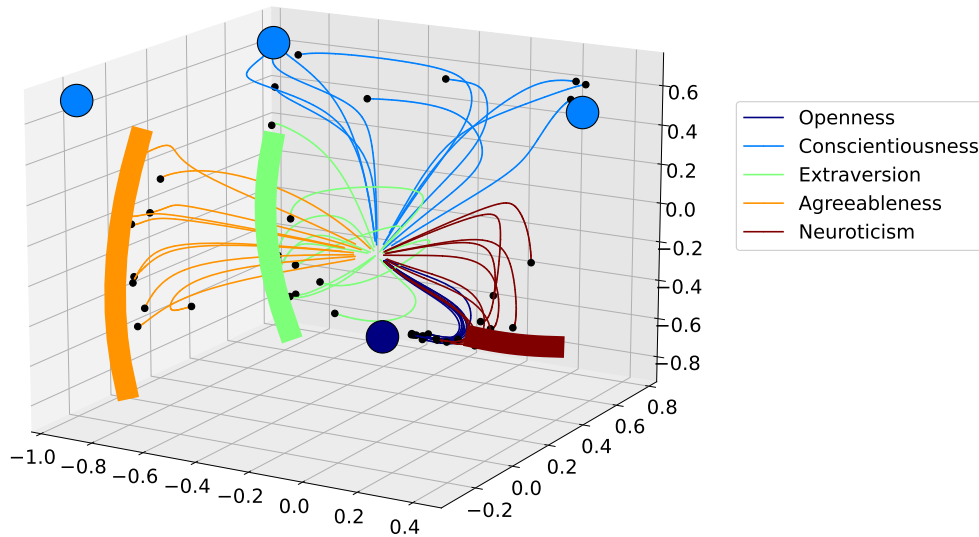


Figure 5.2: The locations of the attractors that govern the dynamics of the state space of our RNN. There exist point and line attractors, labelled according to customers’ dominant personality traits. We show a subset of 100 trajectories, starting from different initial locations, asymptotically converging to their corresponding attractors.

of subclusters along successively less dominant personality traits; lesser personality traits are also drawn to their respective attractors. Intuitively, while people spend differently according to their dominant personality traits, their lesser personality traits still differentiate them within a group of their peers. Therefore, the locations of the attractors are the interpretation of the RNN and allow us to reason about the dynamics of the trajectories.

### 5.3 Affinity-Based RL for Investment Advice

We applied ab-RL to train personal, and interpretable, investment advisors. Five prototypical agents, one for each personality trait, learnt to invest in five different asset classes; each personality trait has unique associations with the different investment classes, as described in Section 4.2. We used publicly available asset index data, shown in Figure 5.3, to calculate the state variables—market indicators over a 24-month moving window—and to calculate the rewards—monthly portfolio returns. The actions of the prototypical agents were the monthly allocations toward asset purchases. For simplicity, assets were never sold or traded. We regularised

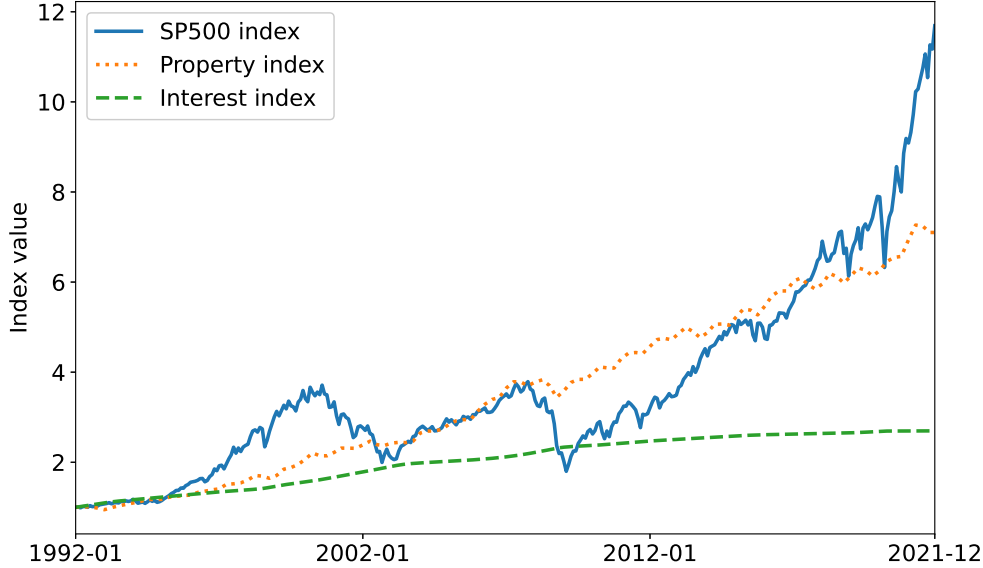


Figure 5.3: Asset pricing data for the S&P500 index, Norwegian property index, and Norwegian interest rate index. The values are relative to their respective values on 1 January 1992. These values are used in the state observations of our RL agents. Taken from Maree and Omlin [58].

these five agents to invest according to the affinities defined in Table 4.3; we scaled these coefficients so that they add up to 1 with values between 0 and 1. We show the regularisation priors in Table 5.1. Each agent thus learnt to maximise monthly returns within the bounds of the preferred action distribution  $\pi_0^a$ ; the openness agent shall, for instance, prefer stocks and luxury items, while the conscientiousness agent shall prefer property and mortgage curtailment. This is in line with the expected preferences of highly open and highly conscientious individuals, respectively; it is the interpretation of the agents’ policies.

Table 5.1: Regularisation priors  $\pi_0^a$  for each agent  $a \in \{\text{openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N)}\}$ . Taken from Maree and Omlin [58].

Investment	$\pi_0^O$	$\pi_0^C$	$\pi_0^E$	$\pi_0^A$	$\pi_0^N$
Savings	0.00	0.07	0.00	0.44	0.64
Property	0.00	0.28	0.00	0.00	0.00
Stocks	0.84	0.00	1.00	0.36	0.12
Luxury	0.16	0.00	0.00	0.00	0.00
Mortgage	0.00	0.65	0.00	0.02	0.24

Figure 5.4 illustrates the five prototypical policies. It is clear that the prototypical agents have learnt to invest according to their defined behaviours. For instance, the openness agent invests mostly in stocks but also in luxury items, while the conscientiousness agent fastidiously invests to reduce the primary debt on mortgages

after having initially invested in property. Note that none of the agents invests in those assets with zero values in their respective priors  $\pi_o^a$ . Therefore, ab-RL not only dictates desired actions, but also prohibits undesired ones.

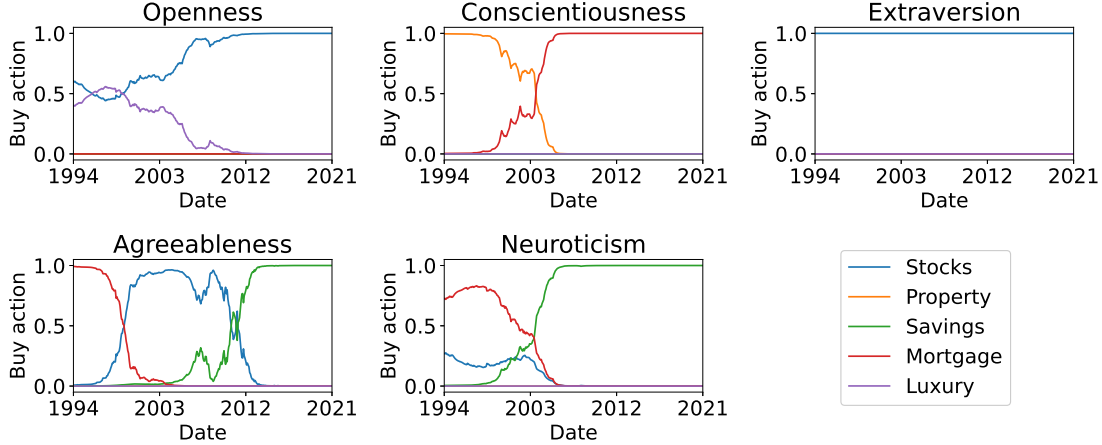


Figure 5.4: Action distributions of the five prototypical agents over a 28-year time period. Each figure represents the investment actions taken by one of the prototypical agents, who each associates with a single personality trait. Each line represents the fractional monthly investment into a class of assets across the time period, e.g. the conscientiousness agent initially invests solely in property and subsequently in mortgage curtailment, while the extraversion agent consistently invests the entire monthly amount in stocks. A declining trend does not indicate selling of assets but rather a reduction of the monthly investment amount; the values on the y-axes are strictly positive indicating our agents never sell assets but rather change their monthly investment distributions. Taken from Maree and Omlin [71].

### 5.3.1 Composition of Prototypical Investment Agents

For the purpose of illustration, we selected four real customers of a major Norwegian bank who demonstrate different spending personalities. For each of these customers, labelled A through D, we used ab-RL to compose our prototypical agents to generate personal investment strategies. We show these customers' personality profiles in Figure 5.5. Customer A has a comparatively balanced personality profile; there is little variation in the (relatively small) values of their personality vector. In contrast, Customer B has high values for openness and neuroticism, Customer C has high values for openness and extraversion, and Customer D has high values for openness, agreeableness, and extraversion. Table 5.2 shows the respective regularisation priors  $\pi_0^c$ ,  $c \in \{A, B, C, D\}$  that we used to train an orchestration agent for each of the four customers. The regularisation priors reflect the personality profiles of the customers; there is little variation in the values of the prior  $\pi_0^A$  for Customer A's agent, while  $\pi_0^B$  has the highest values for openness and neuroticism,  $\pi_0^C$  has the highest values for

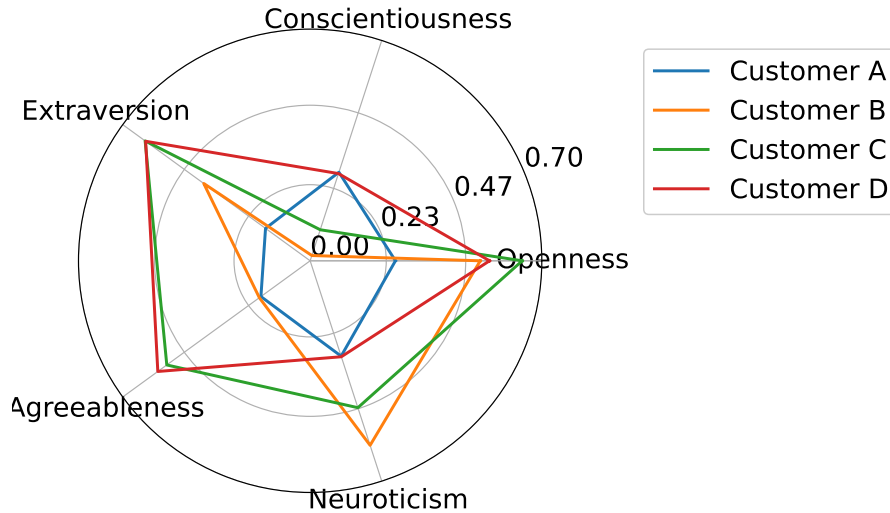


Figure 5.5: The personality vectors representing the personality traits of four real customers. Each coloured line represents a customer and each axis on the radar plot represents a personality trait. The values on the axes are in the range  $[0, 1]$  and represent the customers’ degree of membership in each of the personality traits. These customers were selected to represent a range of personality profiles: Customer A has a balanced profile, Customer B scores high in openness and neuroticism, Customer C scores high in openness and extraversion, and Customer D scores high in openness, agreeableness, and extraversion. Taken from Maree and Omlin [71]

Table 5.2: Regularization priors used during training of the orchestration agents of four customers, named A through D. Each row represents the regularization prior  $\pi_0^c$ ,  $c \in \{A, B, C, D\}$  for one of the orchestration agents. The values are in the range  $[0, 1]$  and add to one for each prior. They represent the fraction of investment amount allocated to each prototypical low-level agent: openness, conscientiousness, extraversion, agreeableness and neuroticism. A higher value indicates a higher weighting of that agent’s strategy. Taken from Maree and Omlin [71].

Prior	Open.	Cons.	Extra.	Agree.	Neur.
$\pi_{0,A}$	0.22	0.24	0.14	0.15	0.25
$\pi_{0,B}$	0.30	0.01	0.23	0.11	0.35
$\pi_{0,C}$	0.27	0.04	0.26	0.23	0.20
$\pi_{0,D}$	0.23	0.12	0.27	0.25	0.13

openness and extraversion, and  $\pi_0^D$  has the largest values for openness, agreeableness, and extraversion.

The orchestration agents used the same state representation as the prototypical agents, but their rewards were a so called “satisfaction index”. We calculated this satisfaction index  $SI = H \cdot (P \cdot C)$  as the dot product between the current asset class holdings in the portfolio  $H$  and a customer’s association with each asset class ( $P \cdot C$ ), where  $P$  is the customer’s personality vector, illustrated in Figure 5.5, and  $C$  is the set of coefficients that associate each asset class with each personality trait, from Table 4.3. The result is a scalar value representing a customer’s affinity for each asset class; it is a measure of the correlation between spending behaviour and investment strategy. This metric is not a fair performance comparison between different customers with different personality profiles; a customer with a perfectly conscientious profile and portfolio will have a different satisfaction index than one with a perfectly extraverted profile and portfolio. It does, however, enable comparison between different policies that compose a strategy for the same customer, which is how we use it.

Figure 5.6 shows the composed investment strategies for the four customers, A through D. Although seemingly similar, there are significant differences between



Figure 5.6: Investment advice from four personal investment agents for four different customer personalities; they are the combined actions of the prototypical agents according to the orchestration agent. Each plot shows the investment advice over time for a single customer, named “Customer A” through “Customer D” from Figure 5.5. Taken from [71].

these strategies. For example, Customer A never exceeded 60% investment in stocks and uniquely invested in property, while Customer D invested up to 90% in stocks. This corresponds to Customer A’s relatively high degree of conscientiousness, i.e.

they prefer lower risk in their portfolio. Customer D, on the other hand, has the greatest risk in their portfolio due to lower investments in property and mortgage curtailment, and more investments in stocks. This corresponds to their low degree of neuroticism, a trait that reduces our appetite for risk. Customer B, compared to Customer C, invested more in savings accounts and less in stocks between 2006 and 2012, which corresponds with Customer B's higher score in neuroticism and lower in agreeableness; in Figure 5.4 we can see that only the neuroticism and agreeableness agents invested in savings, and that the neuroticism agent started such investment much sooner and with higher percentages. We observed that, despite the nuanced differences in investment approaches, the general advice for all customers is consistent with conventional financial advice: younger people with more disposable income may accept more risk for higher returns.

The total investment for the 28-year period was 3.36 million NOK, and there was an initial 2 million NOK property investment with a corresponding 2 million NOK mortgage. This allowed for a variation of individual strategies, e.g. to quickly reduce the principal mortgage debt and thus reduce total interest, or to invest in higher risk and higher reward asset classes, such as stocks. We calculated that the globally optimum strategy—investing solely in stocks—resulted in a maximum return of 27.7 million NOK. Our four orchestration agents achieved returns between 21 and 24 million NOK, which is close to the global optimum. These returns resulted from locally optimum strategies that maximised the correlation between individual customers' spending behaviour and the investment strategy.

### 5.3.2 Time-Variant Composition

To illustrate how these investment strategies can adapt to changing spending behaviours, we created a fictitious customer, Customer E, with a 28-year spending history. The need for a fictitious customer is due to the limited financial history of our real customers, i.e. 6 years. For Customer E, we copied the financial transactions of two distinct real customers: one highly conscientious and the other with a more balanced profile that slightly favoured extraverted spending behaviour. We created Customer E to first exhibit 10 years of conscientious spending behaviour, by copying one year's transactions from the conscientious customer 10 times, then to exhibit 10 years of balanced to extraverted behaviour, by copying one year's transactions from the extraverted customer 10 times, and finally to revert to conscientious spending behaviour, by copying the same transactions from the conscientious customer. Figure 5.7 shows the encoded spending behaviour of Customer E, or the feature trajectory from the state space of the RNN. This trajectory, as expected, first converges



towards the conscientiousness attractor, then changes tack towards the extraverted attractor, and finally returns to the conscientiousness attractor. This demonstrates the interpretability of our feature trajectories: by observing the trajectory, with knowledge of the locations of the attractors, we can reason about the functioning of the model.

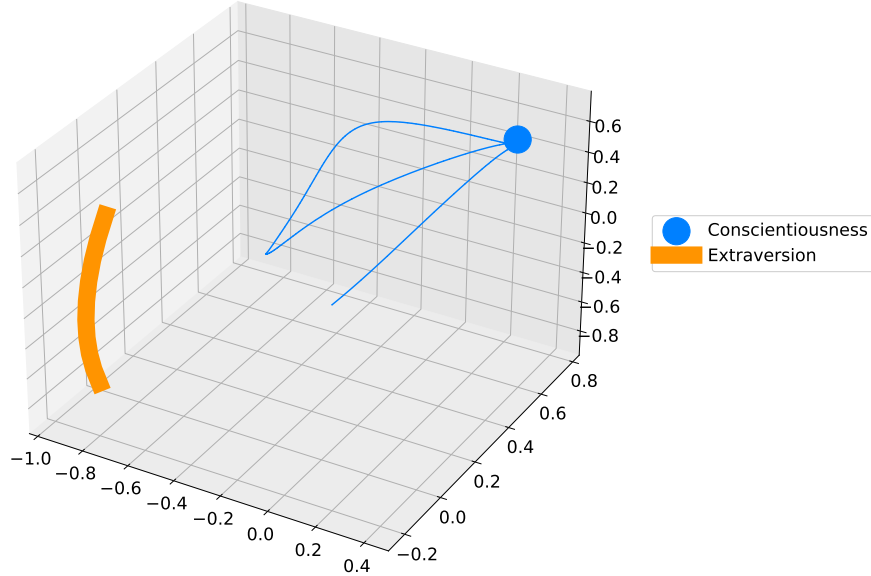
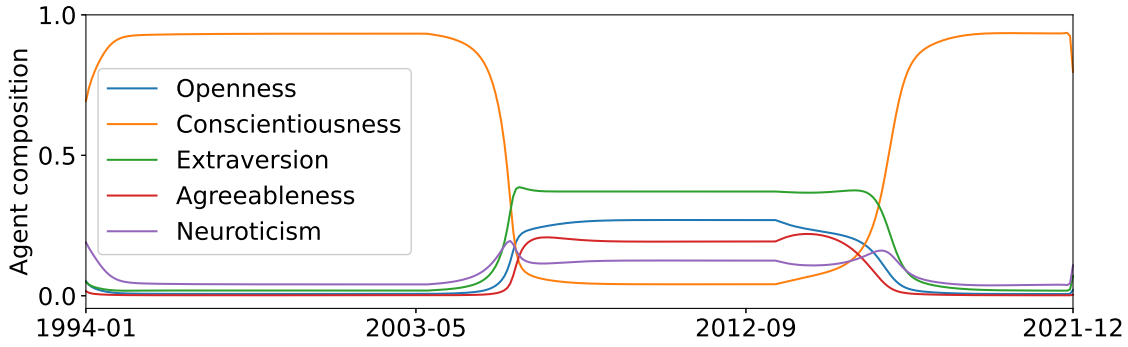
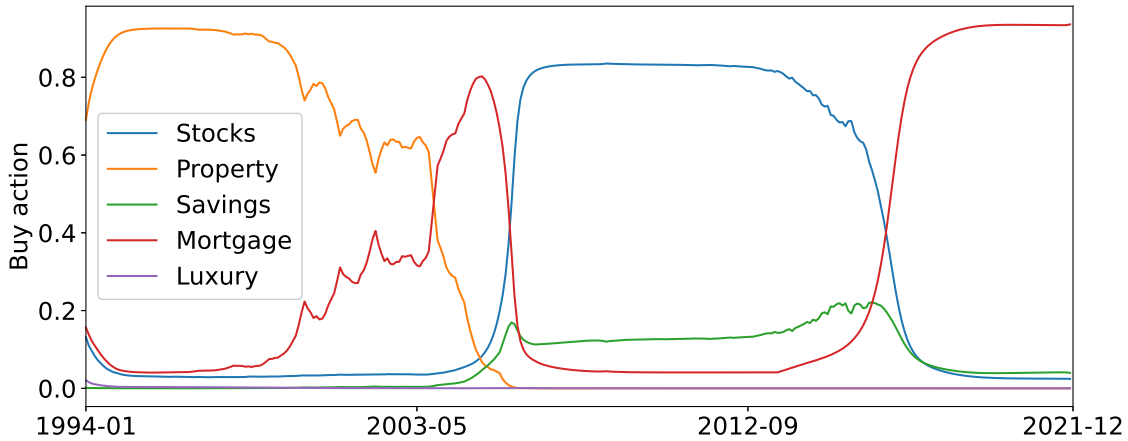


Figure 5.7: The encoded spending behavior for a fictitious customer, Customer E, illustrated as a trajectory in the state space of a RNN. This customer’s financial transactions were such that their spending personality was first predominantly conscientious, then extraverted, and finally conscientious once again. We show the two corresponding attractors and how the customer’s trajectory initially converges on the conscientiousness attractor. When their spending pattern changed, the trajectory moved towards the corresponding new attractor: extraversion. Finally, and before a sufficient time has passed for the trajectory to converge on the new attractor, the spending pattern changes back to conscientiousness and the trajectory once again converges on that attractor. Adapted from Maree and Omlin [71].

To predict the time-dependent orchestration of the five prototypical agents for long-term customers, such as Customer E, we trained orchestration agents for 500 real customers for a six-year period. The actions from these 500 policies were the weights  $\omega_k^j, k \in [1, 5] j \in [1, 500]$  that composed the prototypical agents for each of the 500 customers. We then trained a RNN, with three recurrent nodes, to predict these weights given the sequences of feature trajectories of the 500 customers as input. We used this RNN to predict the time-dependent weights  $\omega_k^E(t)$  given a six-year moving window, in  $t$  time steps, of Customer E’s historical spending behaviour. We show the results in Figure 5.8. This investment strategy highly recommends the conscientiousness policy in the first 10 years, a mostly extraverted policy in the next



(a)



(b)

Figure 5.8: The time-varying composition of prototypical agents for a fictitious customer, Customer E. Customer E displayed conscientious spending behaviour between 1994 and 2004, mostly extraverted behaviour between 2005 and 2015 and conscientious behaviour from 2015 onward. This time-varying spending behaviour is reflected in the weights  $\omega_k^E(t)$ ,  $k \in [\textit{openness}, \textit{conscientiousness}, \textit{extraversion}, \textit{agreeableness}, \textit{neuroticism}]$  assigned in the composition of prototypical agents, shown in (a). The long-term, time-variant investment strategy for Customer E is shown in (b); it initially recommends low-risk asset classes, namely property, but between 2005 and 2015 it recommends stocks and savings accounts, and finally reverts to a conscientious strategy of resolute mortgage curtailment. Adapted from Maree and Omlin [71].

10 years, and finally a conscientious policy. The transitions between policies are not instantaneous, but gradual over a few years. This is important, as financial advice should not be erratic. This composed investment strategy is interpretable from the perspective of changing spending behaviour over time.

### 5.3.3 Explaining Prototypical Agents with Markov Models

Using Markov models, we extracted symbolic explanations for our prototypical agents' policies [61] (refer to Appendix H). We observed the state transitions and

emissions, and directly calculated the transition and emission matrices  $F$  and  $E$ . This involved discretising the state and action spaces, for which we used domain knowledge; market indicators have known threshold values that indicate oversold and overbought conditions, and we divided the continuous action space accordingly into five equally sized bins. The result was a total of 168 potential states, of which only 102 states were visited. Due to the nature of our state space—the market indicators of asset classes—it is not unexpected that certain states never occur. It is, for example, uncommon for one market indicator to suggest oversold conditions while another suggests the opposite, overbought, conditions. We used these trained Markov models to predict, with high fidelity, the actions of the prototypical agents by supplying only the initial state. We show these predictions in Figure 5.9.

Figure 5.10 shows the state transitions for a non-exhaustive subset of states: the first 16 states visited including the initial state. It is infeasible to visualise all state transitions for such a large state space, and we note that transitions to states outside of the selected subset are not shown.

## 5.4 Discussion

Affinity-based RL is a generic paradigm that facilitates the creation of controllable, interpretable strategies through defined traits that each associate with certain actions. These associations define the agents’ affinities that are governed by prior action distributions. The priors instil, through policy regularisation, a global behaviour that prevents agents from selecting certain undesired actions while simultaneously compelling them to select other, desired, actions. Policy regularisation offers certain inherent guarantees, such as a locally optimum solution [15, 27]. Affinity-based RL capitalises on these guarantees by directing the policy to a specific region of the action space. In its current form, ab-RL exacts a global action distribution, but a compelling extension is to define local, or state-dependent, action distributions. Vishwanath et al. [17] envisage an application in ethical AI for such agents; agents could learn to behave according to different virtues—bravery, honor, etc.—depending on the situation or state (refer to Appendix I).

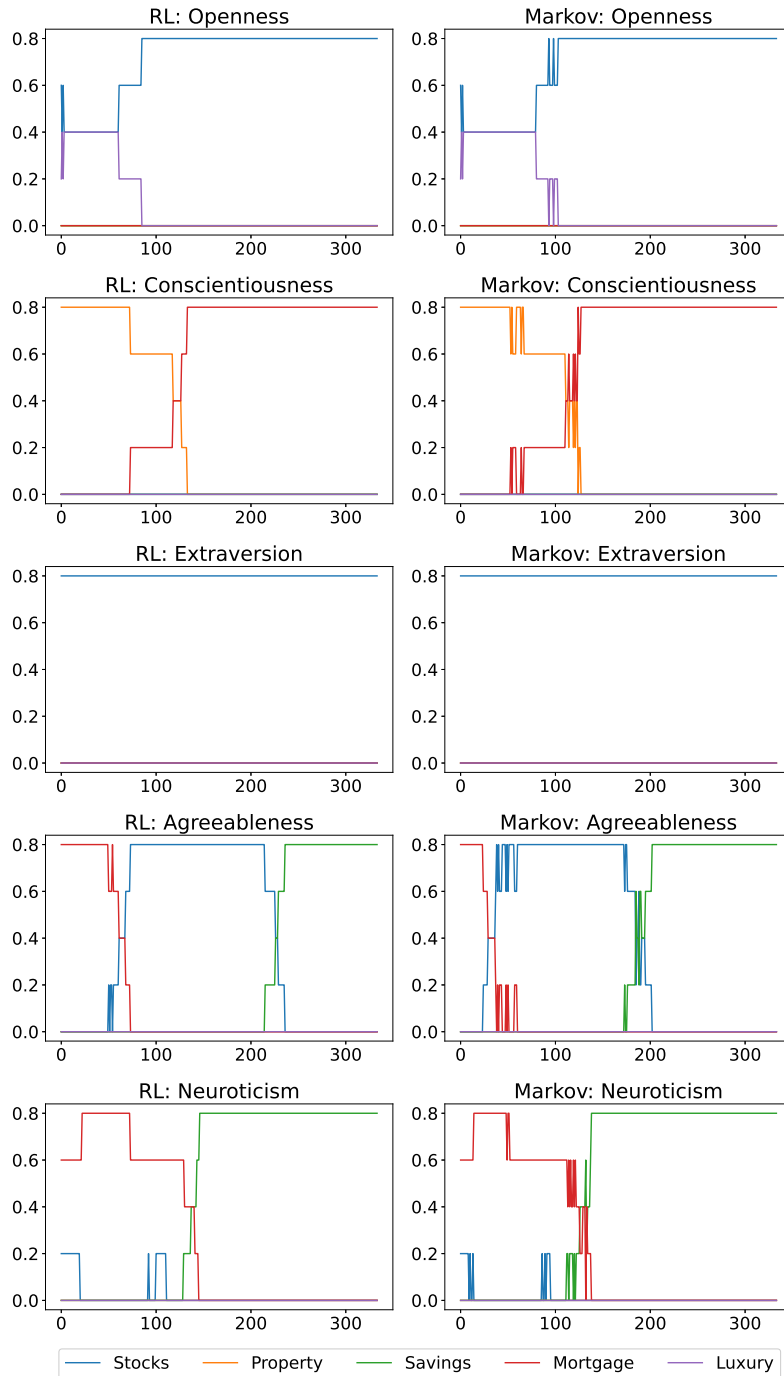


Figure 5.9: A visual comparison between the discretized predictions of five RL agents (on the left) and the five corresponding Markov models (on the right). The single input to the Markov models is the initial state, from which they predict the transition to the next state and the corresponding action by the agent. The Markov models clearly predict the actions with high fidelity.

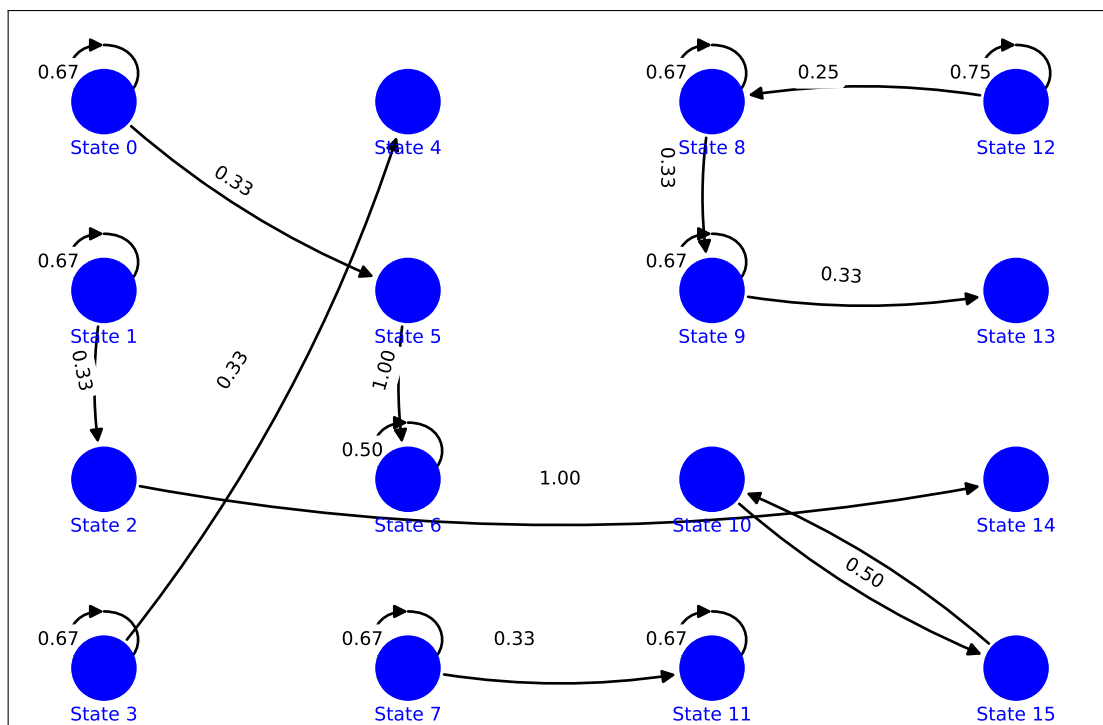


Figure 5.10: A non-exhaustive illustration of the trained Markov model showing state transitions for a subset of states. States are shown as blue circles, and state transitions and their probabilities are shown in black. We show the first 16 states, as visualizing all 102 states is not feasible. Each state represents a set of features with discretized state values. Note that not all state transitions are shown, since the origin or destination state might not be included in this subset. Taken from Maree and Omlin [61].

# Chapter 6

## Conclusions and Future Research Directions

Affinity-based RL allows the interpretation of complex strategies as linear superpositions of prototypical policies. It does this by (1) decomposing tasks along domain-specific traits, (2) defining corresponding action affinities for each trait and instilling desired behaviours into prototypical policies, thus establishing their interpretability, and (3) composing these prototypical policies into linear superpositions. This begs the question whether one can find those linear combinations for an agent that was trained elsewhere on the same task—with or without the use of prototypes. The Kolmogorov–Arnold representation theorem suggests this likelihood. The challenge is to learn the linear coefficients that combine the prototypical strategies, which may vary in time—temporal models, such as RNNs, are key in this pursuit. The relating prototypes may or may not be a complete representation of the underlying domain traits, and this will affect the quality of the interpretations. Such a reverse engineering using ab-RL might yield interesting interpretations of policies that have not been trained using prototypes.

The explainability and interpretability of AI are distinct imperatives in areas that affect human lives, such as finance, health, etc. They pose a non-trivial challenge for RL, which ab-RL naturally solves, and present an opportunity, that ab-RL reveals. Affinity-based RL is consequential in any application where the objective may be a superposition or amalgam of elemental strategies; it reveals that interpretability can be a positive byproduct when solving tasks so complex that they demand simplification through decomposition. It solves the problem of interpretability by intrinsically instilling characteristic behaviours in the policies. The advantages of ab-RL therefore include improved convergence, model understanding by the developer, trust in the model’s decisions, and, in the end, societal acceptance.

We developed our application of personalised investment advice on the commis-

sion of a major Norwegian bank. The resulting system was well received and is considered a highly promising product. We envision an online dashboard where customers may view, and adjust, their calculated personality profiles, their spending behaviour relative to their peers, and their investment recommendations. Other applications of ab-RL include virtuous agents that learn to behave according to human morals, personalised learning and teaching according to students' personalities, treatment of chronic diseases according to patients' health profiles, modelling climate change interventions as social dilemmas, and controlling wind farms to balance peak production and maintenance intervals.

Certain applications, for example the learning of virtuous agents, might require a generalisation of ab-RL from global to local affinities; certain virtues might be desirable only in certain situations, and this may vary over time. Figure 6.1 illustrates the potential for such state-dependent prototypes. Here, a cautious agent steers well clear of dangerous states, regardless of the shortest route. This is a generalisation of an agent with state-independent affinities, for example, the agent in Figure 3.2 that always prefers right turns. It is compelling to compose such state-dependent prototypical agents, e.g. cautious, brave, and honorable agents, to represent complex rational agents that might demonstrate maturing over time through time-dependent local action affinities.

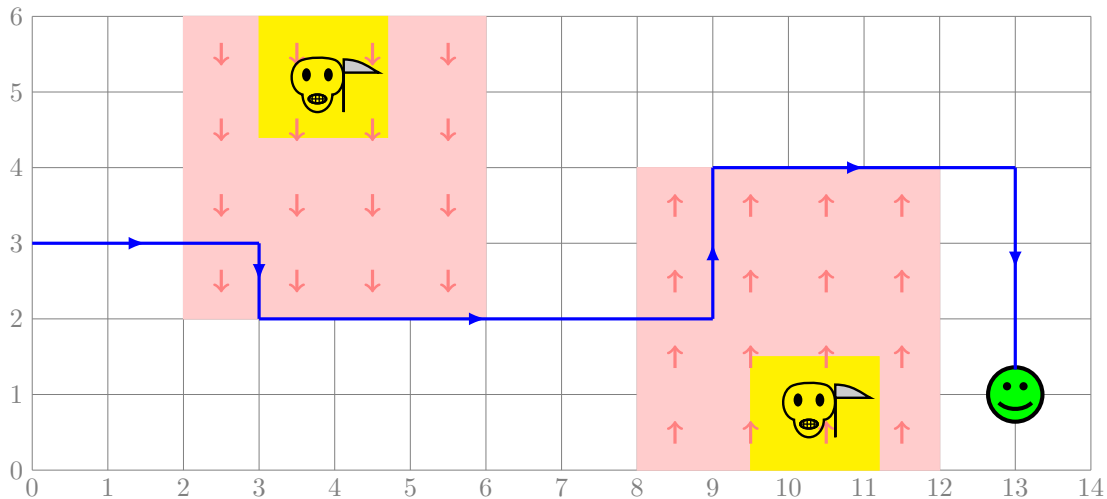


Figure 6.1: The illustration shows the path followed (blue line) by a cautious prototypical agent with a state-specific affinity. The green smiley face is the agent's target destination, and the yellow states are considered dangerous. The red shaded areas indicate states where the agent prefers a specific action (indicated by red arrows). These actions are unique to the current state, or set of states, i.e. moving up or down to avoid danger.

# Bibliography

- [1] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.
- [2] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214(106685): 1–24, 2021.
- [3] Lindsay Wells and Tomasz Bednarz. Explainable AI and reinforcement learning: A systematic review of current approaches and trends. *Frontiers in Artificial Intelligence*, 4:1–48, 2021.
- [4] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. *Machine Learning and Knowledge Extraction. CD-MAKE 2020. Lecture Notes in Computer Science*, 12279, 2020.
- [5] Christian Wirth, Riad Akrouf, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research (JMLR)*, 18(136):1–46, 2017.
- [6] Sobhan Miryoosefi, Kianté Brantley, Hal Daumé, Miroslav Dudík, and Robert E. Schapire. Reinforcement learning with convex constraints. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS)*, pages 1–10, Red Hook, NY, USA, 2019. Curran Associates Inc.
- [7] Waddah Saeed and Christian W Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *arXiv*, 2111.06420: 1–29, 2021.
- [8] Milo Bianchi and Marie Briere. Robo-advising: Less AI and more XAI? *SSRN Electronic Journal*, pages 1–29, 2021.



- [9] Yanou Ramon, R A Farrokhnia, Sandra C Matz, and David Martens. Explainable AI for psychological profiling from behavioral data: An application to big five personality predictions from financial transaction records. *Information (Basel)*, 12(12):518–547, 2021.
- [10] Warren Freeborough and Terence van Zyl. Investigating explainability methods in recurrent neural network architectures for financial time series data. *Applied Sciences (Basel)*, 12(3):1427–1442, 2022.
- [11] Giuseppe Cascarino, Mirko Moscatelli, and Fabio Parlapiano. Explainable artificial intelligence: Interpreting default forecasting models based on machine learning. *Questioni di Economia e Finanza*, 1(674):1–38, 2022.
- [12] Miseon Han and Jeongtae Kim. Joint banknote recognition and counterfeit detection using explainable artificial intelligence. *Sensors*, 19(16):1–18, 2019.
- [13] Chanyuan Abigail Zhang, Soohyun Cho, and Miklos Vasarhelyi. Explainable Artificial Intelligence (XAI) in auditing: A framework and research needs. *SSRN Electronic Journal*, pages 1–58, 2021.
- [14] Solon Barocas and Andrew D. Selbst. Big data’s disparate impact. *California Law Review*, 104(671):671–732, 2016.
- [15] Tuomas Haarnoja, Haoran Tang, P. Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1352–1361, 2017.
- [16] Alexandre Galashov, Siddhant Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojtek M. Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in KL-regularized RL. In *International Conference on Learning Representations (ICLR)*, pages 1–25, New Orleans, Louisiana, United States, 2019.
- [17] Ajay Vishwanath, Einar Duenger Bøhn, Ole-Christoffer Granmo, Charl Mæree, and Christian Omlin. Towards artificial virtuous agents: Games, dilemmas and machine learning. *AI and Ethics (In Print)*, 2(4):1–19, 2022.
- [18] Richard Bellman. A Markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.
- [19] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

- [20] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv*, 1509.02971:1–14, 2019.
- [21] Junior C. Jesus, Jair A. Bottega, Marco A. S. L. Cuadros, and Daniel F. T. Gamarra. Deep deterministic policy gradient for navigation of mobile robots in simulated environments. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 362–367, 2019.
- [22] Guang Yang, Feng Zhang, Cheng Gong, and Shiwen Zhang. Application of a deep deterministic policy gradient algorithm for energy-aimed timetable rescheduling problem. *Energies*, 12(18):1–19, 2019.
- [23] Ayman Chaouki, Stephen Hardiman, Christian Schmidt, Emmanuel Sérié, and Joachim de Lataillade. Deep deterministic portfolio optimization. *The Journal of Finance and Data Science*, 6:16–30, 2020.
- [24] Charl Maree and Christian W. Omlin. Balancing profit, risk, and sustainability for portfolio management. In *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFER)*, pages 1–8, 2022.
- [25] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *arXiv*, 1908.06976, 2019.
- [26] Alain Andres, Esther Villar-Rodriguez, and Javier Del Ser. Collaborative training of heterogeneous reinforcement learning agents in environments with sparse rewards: What and when to share? *arXiv*, 2202.12174, 2022.
- [27] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Remi Munos, and Matthieu Geist. Leverage the average: An analysis of KL regularization in reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 33, pages 12163–12174. Curran Associates, 2020.
- [28] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. In *Journal of Machine Learning Research (JMLR)*, volume 18, pages 1–51, 2018.
- [29] Zengyi Qin, Yuxiao Chen, and Chuchu Fan. Density constrained reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume PMLR 139, 2021.
- [30] Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer, Berlin, Heidelberg, 2010.

- [31] Usha Goswami. Inductive and deductive reasoning. *The Wiley-Blackwell handbook of childhood cognitive development*, pages 399–419, 2011.
- [32] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5045–5054, 2018.
- [33] Guiliang Liu, Oliver Schulte, Wang Zhu, and Qingcan Li. Toward interpretable deep reinforcement learning with linear model u-trees. In *Machine Learning and Knowledge Discovery in Databases*, pages 14–429, 2018.
- [34] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-François Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 108:379–392, 2018.
- [35] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. *arXiv*, 1711(00138v5):1–10, 2018.
- [36] Pedro Sequeira, Eric Yeh, and Melinda T. Gervasio. Interestingness elements for explainable reinforcement learning through introspection. *IUI Workshops*, pages 1–7, 2019.
- [37] Ofra Amir, Finale Doshi-Velez, and David Sarne. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems*, 33:1–17, 2019.
- [38] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv*, 1609.05518, 2016.
- [39] Marta Garnelo and Murray Shanahan. Reconciling deep learning with symbolic artificial intelligence: Representing objects and relations. *Current Opinion in Behavioral Sciences*, 29:17–23, 10 2019.
- [40] Artur d’Avila Garcez, Aimore Resende Riquetti Dutra, and Eduardo Alonso. Towards symbolic reinforcement learning with common sense. *arXiv*, 1804.08597, 2018.
- [41] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. *arXiv*, 1905.10958v2, 2019.
- [42] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley Q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(5):7285–7292, 2020.

- [43] Régis Riveret, Yang Gao, Guido Governatori, Antonino Rotolo, Jeremy V. Pitt, and Giovanni Sartor. A probabilistic argumentation framework for reinforcement learning agents. *Autonomous Agents and Multi-Agent Systems*, 33: 216–274, 2019.
- [44] Geraud Nangue Tasse, Steven James, and Benjamin Rosman. A Boolean task algebra for reinforcement learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.
- [45] Harm van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid reward architecture for reinforcement learning. *arXiv*, 1706.04208, 2017.
- [46] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. *International Joint Conference on Artificial Intelligence. A Workshop on Explainable Artificial Intelligence*, 2019.
- [47] Luca Marzari, Ameya Pore, Diego Dall’Alba, Gerardo Aragon-Camarasa, Alessandro Farinelli, and Paolo Fiorini. Towards hierarchical task decomposition using deep reinforcement learning for pick and place subtasks. *arXiv*, 2102.04022, 2021.
- [48] Benjamin Beyret, Ali Shafti, and A. Aldo Faisal. Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 5014–5019, 2019.
- [49] Daoming Lyu, Fangkai Yang, Bo Liu, and Steven M. Gustafson. SDRL: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. *AAAI Conference on Artificial Intelligence*, 3(1):1–9, 2019.
- [50] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems (NIPS)*, 30(1):1–12, 2017.
- [51] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

- [52] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *arXiv*, 1602.01783, 2016.
- [53] Johannes Schmidt-Hieber. The Kolmogorov–Arnold representation theorem revisited. *Neural Networks*, 137:1–8, 2021.
- [54] Shreeram S. Abhyankar. Hilbert’s thirteenth problem. In *Mathematics Subject Classification*, Mathematics Department, Purdue University, West Lafayette, IN 47907, USA, 1985.
- [55] John Lu, Sunanda Dissanayake, Nelson Castillo, and Kristine Williams. Safety evaluation of right turns followed by u-turns as an alternative to direct left turns - conflict analysis. Technical report, CUTR Research Reports, 2001.
- [56] Charl Maree and Christian W. Omlin. Reinforcement learning your way: Agent characterization through policy regularization. *AI*, 3(2):250–259, 2022.
- [57] Simone Parisi, Voot Tangkaratt, Jan Peters, and Mohammad Khan. TD-regularized actor-critic methods. *Machine Learning*, 108(8):1–35, 2019.
- [58] Charl Maree and Christian W. Omlin. Can interpretable reinforcement learning manage prosperity your way? *AI*, 3(2):526–537, 2022.
- [59] Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.
- [60] Fanny Yang, Sivaraman Balakrishnan, and Martin J. Wainwright. Statistical and computational guarantees for the Baum-Welch algorithm. *Journal of Machine Learning Research*, 18(125):1–53, 2017.
- [61] Charl Maree and Christian W. Omlin. Symbolic explanation of affinity-based reinforcement learning agents with Markov models. *arXiv*, 2208.12627:1–11, 2022.
- [62] Sandra. C. Matz, Michal Kosinski, Gideon Nave, and David. J. Stillwell. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the National Academy of Sciences*, 114(48):12714–12719, 2017.
- [63] Niru Maheswaranathan, Alex H. Williams, Matthew D. Golub, S. Ganguli, and David Sussillo. Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in neural information processing systems (NIPS)*, 32(1):15696–15705, 2019.

- [64] Sandra Matz, Joe Gladstone, and David Stillwell. Money buys happiness when spending fits our personality. *Psychological science*, 27(5):1–11, 2016.
- [65] Muhammad Zubair Tauni, Zia-ur-Rehman Rao, Hongxing Fang, Sultan Sikan-dar Mirza, Zulfiqar Ali Memon, and Khalil Jebran. Do investor’s Big Five personality traits influence the association between information acquisition and stock trading behavior? *China Finance Review International*, 7(4):450–477, 2017.
- [66] Natkamon Tovanich, Simone Centellegher, Nacéra Bennacer Seghouani, Joe Gladstone, Sandra Matz, and Bruno Lepri. Inferring psychological traits from spending categories and dynamic consumption patterns. *EPJ Data Science*, 10(24):1–23, 2021.
- [67] Charl Maree and Christian W. Omlin. Clustering in recurrent neural networks for micro-segmentation using spending personality. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–5, 2021.
- [68] Andrea Ceni, Peter Ashwin, and Lorenzo Livi. Interpreting recurrent neural networks behaviour via excitable network attractors. *Cognitive Computation*, 12(2):330–356, 2019.
- [69] Charl Maree and Christian W. Omlin. Understanding spending behavior: Re-current neural network explanation and interpretation. In *IEEE Computational Intelligence for Financial Engineering and Economics*, pages 1–7, 2022.
- [70] Knight Frank Company. Knight Frank luxury investment index, 2022. <https://www.knightfrank.com/wealthreport/luxury-investment-trends-predictions/>, Accessed on 2022-05-27.
- [71] Charl Maree and Christian W. Omlin. Reinforcement learning with intrinsic affinity for personalized prosperity management. *Digital Finance*, 4(3):241–262, 2022.
- [72] Charl Maree, Jan Erik Modal, and Christian W. Omlin. Towards responsible AI for financial transactions. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 16–21, 2020.



# Appendix A

## Towards Responsible AI for Financial Transactions

This paper has been published as:

C. Maree, J. E. Modal and C. W. Omlin, “Towards Responsible AI for Financial Transactions”, *IEEE Symposium Series on Computational Intelligence (SSCI)*, **2020**, pp. 16–21, doi: 10.1109/SSCI47803.2020.9308456.

Copyright © 2020 IEEE



# Towards Responsible AI for Financial Transactions

Charl Maree<sup>1</sup>  
Center for AI Research  
University of Agder  
Grimstad, Norway  
charl.maree@uia.no

Jan Erik Modal  
SpareBank1 Development  
SpareBank1 Alliance  
Oslo, Norway  
jan.erik.modal@sparebank1.no

Christian W. Omlin  
Center for AI Research  
University of Agder  
Grimstad, Norway  
christian.omlin@uia.no

**Abstract**—*The application of AI in finance is increasingly dependent on the principles of responsible AI. These principles – explainability, fairness, privacy, accountability, transparency and soundness form the basis for trust in future AI systems. In this empirical study, we address the first principle by providing an explanation for a deep neural network that is trained on a mixture of numerical, categorical and textual inputs for financial transaction classification. The explanation is achieved through (1) a feature importance analysis using Shapley additive explanations (SHAP) and (2) a hybrid approach of text clustering and decision tree classifiers. We then test the robustness of the model by exposing it to a targeted evasion attack, leveraging the knowledge we gained about the model through the extracted explanation.*

**Keywords**—*AI in Finance, Explainable AI, Feature Saliency, SHAP, Text Clustering, Rule Extraction, Decision Trees*

## I. INTRODUCTION

AI is becoming increasingly more omnipresent in the financial industry, with applications in customer interaction, investor services, fraud detection, customer relationship management and anti money laundering [1]. There exists enormous business potential for advanced analytics and economic modelling. Customer relations can be improved through innovative services in the form of digital financial advisors or personal assistants. Personal assistants such as Google Alexa, Apple Siri and Google Assistant have been developed for many applications.

Currently, chatbots are the primary interface for digital assistants in finance. In the future, digital financial assistants will move beyond question answering and play a more active role in wealth management, smart payment solutions and credit and insurance management [1].

The need for the responsible application of AI in finance has been highlighted [1] [2] [3], and there is increased interest in responsible AI [4].

This study considers a financial transaction classification model. Electronic financial transactions are classified into categories, such as “groceries”, “transportation”, “savings”, etc. The transactions are retrieved from the database of a major Norwegian bank and offer a good representation of the actual spending habits of customers. In Scandinavia, cash represents less than 5% of the total money supply [5] and is declining [6]. Electronic transactions therefore capture a significant portion of customer spending. The classifications made by the model of interest will, in the future, be used to develop a series of value adding products for customers, with the end goal of developing a digital financial advisor. As the basis for future work, it is important that the transaction classifier be implemented in accordance with the principles of responsible AI. In this study, we address one of the core principles of responsible AI, namely explainability.

The aim of this empirical study is therefore to (1) identify the salient features of the transaction classification model and (2) extract an explanation for the function of the model, i.e. the rules that govern the model. We also illustrate vulnerability of the current financial transaction model to perturbations of the salient inputs. We achieve these goals in a hybrid approach where (1) we determine the feature importance using Shapley additive explanations (SHAP), (2) we generate explanations using a combination of clustering and decision trees and (3) we show model susceptibility to adversarial examples, leveraging knowledge from the explanation.

In Section II, we discuss the concept of responsible AI. We then briefly review related work in Section III. We describe our transaction classification model in Section IV and we present the theory behind the methods used for extracting an explanation. Section V describes the methodology for our experiments, and we discuss the results in Section VI. The paper closes with a summary, conclusions and directions for future research in Section VII.

## II. RESPONSIBLE AI

Responsible AI provides a framework that focuses on ensuring the ethical, transparent and accountable use of AI technologies in a manner consistent with user expectations, societal laws and norms. It can guard against the use of biased data or algorithms, ensure that automated decisions are justified and explainable, and ensure user trust and individual privacy.

The principles of responsible AI can generally be summarized as fairness, privacy, accountability, transparency and soundness; however, no consensus exists on either a definition or measures for their quantification [4].

ML algorithms tend to adopt the bias present in the training data. This could translate into discrimination, e.g. credit rating according to postal codes [7], which violates the standards of fairness. AI systems can potentially use personal information in ways that intrude on individual privacy [8] by collecting and relating data that then becomes a commodity beyond the individual’s knowledge or control. Accountability ensures that the system operator can be held liable for any adverse effects or consequences of the actions of AI systems; it does not necessarily remove bias. The imperative of AI transparency demands explainability and interpretability of AI systems, as well as data provenance. Explainability provides an accurate proxy or symbolic representation of the AI system whereas interpretability explains a model’s predictions in human understandable terms, e.g. in relation to the input features. Explainability does not automatically imply interpretability [4]. Trustworthy AI systems must be reliable and accurate, behave predictably, and operate within in the boundaries of applicable rules and regulations. This also

<sup>1</sup> Strategy Innovation and Development, SpareBank 1 SR-Bank ASA, Norway.

implies robustness and security against attacks such as poisoning or evasion, as demonstrated in [9]. These core principles of responsible AI must be equally weighted in any responsible AI application.

### III. RELATED WORK

Interpretable models require mitigation of their complexity; an explanation of an AI system may have high fidelity and accuracy, but it may be incomprehensible to humans. There is a common perception about the existence of a trade-off between model interpretability and performance [4]; the work reported in [10] addresses this issue. It unifies six existing methods; which lack certain desirable properties: (1) local accuracy, which requires the explanation model to at least match the output of the target model for some simplified input; (2) missingness, which requires features with zero values to have no attributed impact; (3) consistency, which states that if a model changes such that some simplified input's contribution does not decrease, then that input's attribution should increase or remain the same, irrespective of the other inputs.

The six unified methods are (1) local interpretable model explanations (LIME), which explains model predictions based on local approximations of the model around a given instance; (2) deep learning important features (DeepLIFT), which measures the change in model output resulting from changing a given input value to a reference value; (3) layer-wise relevance propagation, which estimates feature relevance from the changes prediction similar to DeepLIFT but uses a different underlying mechanism; (4) Shapley regression values, which calculate feature importance for linear models by retraining the model on different subsets of the features; (5) Shapley sampling values, which approximate the effect of removing a variable from the model by integrating over samples from the training set and (6) quantitative input influence, which addresses more than just feature importance, but that independently proposes sampling approximation which is nearly identical to Shapley values.

In general, calculating the exact SHAP values is a computationally impractical problem. SHAP unifies the insights from methods 1-6 to approximate them (see Section IV.B). In [11], the authors apply SHAP in order to explain the predictions of a non-linear model on a financial time-series. They reveal the salient features and show which features are responsible for predicting a given class of output. They show how SHAP values can be used to improve prediction accuracy by assessing the usefulness of adding additional data.

Once we have identified salient features, we intend to simplify the input space by means of clustering. In [12], the authors identify salient features, then use the most important feature to reduce model complexity through clustering of the input space; they then fit a unique decision tree on each cluster. The resulting small decisions trees are more compact and thus more interpretable than a single larger tree. In [13], a dataset is clustered in order to improve the performance of a decision tree classifier. The idea is that many smaller classifiers are more elastic in terms of underlying algorithms and parameters, compared to a single, larger classifier. The authors report a 40% improvement in classification performance using this method.

### IV. CLASSIFICATION MODEL AND EXPLANATION

Our target system is a transaction classification system which is currently in production and receives between 10 and 1500 requests per second. A typical request has about 100 transactions and processing time for requests increases linearly with the number of transactions. Processing time is typically between 2ms and 50ms.

In this section, we discuss the features used in the target model and give a short overview of the target model. We then introduce the methods extracting an explanation.

#### A. Feature Encoding and Target Model

The features in the dataset include categorical, numerical and text attributes. The target model is a series of two opaque models: a word2vec encoder followed by a deep neural network (DNN). In the first model, the transaction text is encoded into a vector representation

$$X_t = \{X_t^i\}, i \in \{1, \dots, n\} \subset N \quad (1)$$

where  $N$  is the number of features in the feature space,  $n$  is the dimensionality of the vector representation of the text, i.e. the product of the size of embedding vector  $k$ , and the number of words in the text  $l$ , i.e.  $n = k \times l$ . This vector is concatenated with one-hot encodings of the transaction code  $X_c$  and day of week  $X_d$ , normalized transaction amount  $X_a$  and customer age  $X_g$  as well as binary series representing whether the transaction amount is negative or positive (payment vs deposit)  $X_d$  and whether the amount includes cents  $X_e$ . This concatenated dataset  $X$ , which is sent into a DNN is formally represented by:

$$X = \{X_t, X_c, X_d, X_a, X_g, X_d, X_e\} \quad (2)$$

The model is a classification net, producing a probability distribution  $Y_i \in Y$ ,  $i \in \{1, \dots, m\}$  where  $m$  is the number of output classes.

The training set was labelled using a mixed technique of defined rules and manual labelling. The rules did not accurately classify all transaction; misclassified transactions had to be hand labelled.

#### B. Salient Feature Extraction using SHAP

The selection of salient features i.e. features containing high predictive information, is imperative for the development of machine learning models with high performance, particularly when it involves high-dimensional feature spaces. An ad-hoc heuristic trains and tests models with features omitted one at a time. Shapley additive explanations (SHAP) [10] offers an alternative, mathematically sound and parsimonious approach to salient feature extraction. In [14] the authors demonstrate that SHAP appropriately adjusts feature salience ratings when features are replaced one at a time with random noise.

SHAP is based on the collaborative game theory method, Shapley values [15]. It clarifies the prediction of an instance  $x \in X$ , where  $X$  is the set of all instances, by computing the contribution of each input feature  $x_i \in x$ ,  $i \in \{1, \dots, N\}$  where  $N$  is the number of features in the dataset. SHAP values assign weights to each feature cluster, where a feature cluster can be either a single feature, e.g. in numeric data, or a group of features, e.g. several words in a sentence. SHAP uses these weights in an additive linear model to explain the overall

contribution of all features, thus elegantly blending elements from Shapely values [15], LIME [16] and others.

In [10], the authors define a given explanation model  $g$  as

$$g(z') = \phi_0 + \sum_{j=1}^N \phi_j z'_j \quad (3)$$

where  $z' \in \{0,1\}^N$  is the feature space vector indicating the presence of each feature,  $N$  is the size of the feature space and SHAP values  $\phi_j \in \mathbb{R}$  is the individual feature contribution for a feature  $j$ . The feature space refers to a simplified feature space that maps to the original feature space through a mapping function  $z = h_z(z')$ . The individual feature contributions  $\phi_j \in \mathbb{R}$  are estimated using the collaborative game theory approach Shapley [15].

Shapley explores a game where the prediction of a model  $f(x)$  is seen as the result, or payout, of the game. The individual features  $x_i \in x$  are the players. The goal is to determine the contribution that each player has to the payout. Shapley determines how to fairly distribute the payout among the players through comparison of the model outputs for different coalitions of feature values. Feature coalitions are made by randomly sampling values from the feature space, i.e. a coalition is a fictitious instance  $x' \notin X$ , where feature values of the instance  $x'_i \in x$  are drawn randomly from the feature space. The Shapley value of a feature is defined as the average change in the prediction  $\Delta \hat{f}(x) = \hat{f}(x') - \hat{f}(x'')$  that a coalition  $x'$  receives when a new feature value  $x'_i \in x$  joins the coalition.

### C. Text Clustering using DBSCAN

Text is typically clustered using a spatial clustering algorithm, such as the density based spatial clustering algorithm with noise (DBSCAN) [17], [18]. It starts with a random instance and identifies all its nearest neighbors. Proximity to other instances is determined through a given distance measure, e.g. Euclidean, Hamming, Cosine, etc. If a point has a minimum of  $minPts$  neighbors within a distance of  $\epsilon$ , then a new cluster is defined. The algorithm will also identify outliers that do not fall in any cluster as noise.

When text is represented as word vectors, through e.g. a word2vec encoder, the similarity between two sentences corresponds to the distance between the vectors. This is generally quantified as the cosine of the angle between the vectors [19] i.e. the cosine similarity. Given two sentences, the cosine similarity is defined as:

$$sim_c(t_i, t_j) = \frac{t_i \cdot t_j}{|t_i| \times |t_j|} \quad (4)$$

Where  $t_i, t_j \in T$ , are  $n$ -dimensional vectors in the term set  $T = \{t_1, \dots, t_n\}$  and  $sim_c \in [0,1]$ . When two terms are identical, the cosine similarity is 1 i.e.  $sim_c(t_k, t_l) = 1, \forall t_k = t_l$ .

DBSCAN can therefore be used with cosine similarity as a clustering method for texts.

## V. EMPIRICAL METHODOLOGY

### A. Data

Throughout this study, we used an initial dataset of roughly 10 million financial transactions. These transactions

were labelled using the target model and resampled without replacement to provide a more uniform representation of the labelled classes. The final dataset,  $X$ , had a cardinality of roughly 5 million transactions, i.e.  $|X| \cong 5\,000\,000$ .

### B. Explanation by Decision Trees

Global surrogate modelling is a well-documented approach to model explainability [4]. In this study, we trained both a single decision tree and a random forest as global surrogates to explain the model. We used a random sample of 10% of the total dataset (about half a million transactions) for training,  $X_{train} \subset X \wedge |X_{train}| = 0.1 \times |X|$ , while testing was done on a randomly sampled set of 100 000 transactions,  $X_{test} \subset X \wedge X_{test} \notin X_{train} \wedge |X_{test}| = 100\,000$ .

We used these train and test sets to fit a decision tree classifier and a random forest classifier with 50 individual trees. The performance and human understandability of the tree and forest were used as a baseline to compare with a hybrid clustering / decision tree approach discussed below.

### C. Feature Importance through SHAP Analysis

We estimate the feature importance using SHAP [10]. The feature importance,  $\phi_i$  was estimated for each input feature  $x_i \in \{1, \dots, N\}$  where  $N$  is the total number of encoded features. Note that due to encoding,  $N > 7$  where 7 is the number of original features.

Equations (1) and (2) illustrate how the features are prepared, with equation (1) referring to the word vectors for the transaction text. SHAP values provide an estimate of the importance of individual features,  $\phi_i \rightarrow X_t^i$ ; however, this is not useful when the feature of interest is a superfeature:  $X_t = \{X_t^i, i \in \{1, \dots, n\}\}$ . In order to derive the importance of the superfeature  $X_t$ , we aggregate the SHAP values through addition [10]:

$$\phi_t = \sum_1^n \phi_i \quad (5)$$

### D. Explanation through Clustering and Decision Trees

Having identified the most important feature, we clustered the data according to this feature. In Section VI, we show that the most important feature in classification is the transaction text  $X_t$ ; we therefore used the DBSCAN algorithm with cosine similarity as the distance measure. We trained a set of  $m$  superclusters,  $c_i \in C, i \in \{1, \dots, m\}$ , where  $m$  is the number of classes in the output  $y_i \in Y, i \in \{1, \dots, m\}$ .

From these superclusters, we considered the individual words from the texts contained in each cluster. We created a list of keywords  $k_i \in K$  for each supercluster  $i$  by extracting unique words from each cluster. Stop words such as place and street names were removed from the keyword lists. Formally,

$$k_i \in K \wedge k_i \cap k_j = \emptyset \quad (6)$$

$$i, j \in \{1, \dots, m\} \wedge i \neq j$$

The keywords were used as rules that associate a given transaction text with a given supercluster. For any given transaction text  $t$ , each word in the text  $w \in t$  was given the opportunity to vote for a supercluster  $c$ ; we considered the keyword list for each supercluster; if a word  $w$  appears in the keywords list  $k_i$ , the word voted for supercluster  $c_i$ . The votes

for all words were tallied and the supercluster was selected through majority vote. If no supercluster was found, i.e. no words appear in the keyword list, a default supercluster representing the class “other” was selected. We used shallow decision trees to filter out those instances that did not belong to the homogeneous majority. This is similar to the approach in [12] and [13]; we intended to simplify the final explanation while simultaneously attaining improved accuracy compared to a single large classifier.

#### E. Model Robustness against Evasion Attacks

In order to test the robustness of the model, we subjected the model to a targeted evasion attack, leveraging the newfound knowledge about the model. A successful adversarial attack therefore suggests not only a vulnerability in the model, but also a working knowledge of the model by the attacker.

The adversarial examples were generated by slightly perturbing existing instances, along the feature of highest importance, i.e. where the impact would be greatest. The perturbations therefore targeted the transaction text,  $X_t$ . The perturbed set of adversarial examples  $X_{pert} \in X$  is therefore defined by:

$$x' = \{x_t', x_c, x_d, x_a, x_g, x_d, x_e\} \quad (7)$$

$$x \in X \wedge x' \in X_{pert}$$

Words from the texts were selected by matching the words with the keyword dictionary,  $K$ . If a word appeared in one of the keyword lists  $k_i \in K$ , then that word was replaced by a word from another list  $k_j \in K$ , where  $i \neq j \wedge i, j \in \{1, \dots, m\}$ .

## VI. RESULTS

The labelled set of transactions was divided into training (80%), validation (10%) and test (10%) sets. The trained model achieved a mean accuracy of 98.2%, with a 95% confidence interval of 0.04% in 20 experiments.

As a baseline to an explanation, we trained a decision tree and a random forest as surrogate models on data labelled by the DNN. The decision tree achieved an accuracy of 95.35% (95% confidence interval of 0.02%), while the random forest (with 50 estimators) achieved an accuracy of 96.2% (95% confidence interval of 0.02%). Both the single decision tree and the random forest had in excess of 50 000 nodes. Even though decision trees inherently explained the rules they have derived, they clearly do not provide interpretability in this instance.

#### A. Feature Importance and Model Explanation

The results from the SHAP feature importance evaluation are clear evidence of the model’s bias towards the text features. As seen in Fig. 1, the transaction text is largely responsible for the predictions. This is consistent with the importance of transaction text for the partial labelling of the original dataset.

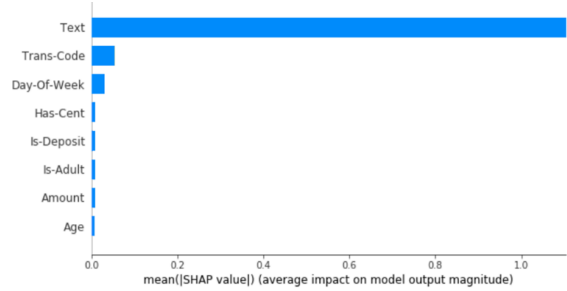


Fig. 1 The feature importance estimation by SHAP analysis shows that the transaction text is the most important feature for the transaction classification.

Knowing that the text is the most important feature for model classification is not an adequate explanation of the functioning of the model. To determine how the model uses the text, we used its vector representation in a clustering analysis; we used DBSCAN with the distance parameter  $\epsilon = 0.07$ . The intent was to train tight clusters. The result was a set of clusters with high homogeneity (95%) and a low percentage of noise (2%), with a total of 12 734 clusters. We then grouped the clusters using the labelled training data into  $m$  superclusters. Fig. 2 shows a 2-dimensional representation of the supercluster for transactions relating to alcohol.

2D Illustration of cosine similarity in clusters for class: Alcohol

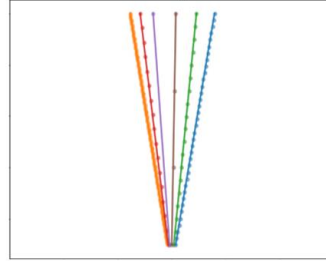


Fig. 2. Text vectors are clustered and grouped into superclusters. A 2-dimensional projection of the supercluster “Alcohol” is shown with each of the clusters containing several instances from the training set. The angles of the clusters shown in this plot are equal to the angles in the word2vec text embedding dimension.

Finally, we fit a small, interpretable decision tree to each supercluster with less than 100% homogeneity; the shallow decision tree provides the final separation and explanation. An example tree is shown in Fig. 3, for cluster number 10 relating to expenditure on kindergartens.

In Fig. 3, the tree distinguishes between transactions relating to kindergarten and those relating to property management. The transaction code “014” is the most important feature in this classification, while the amount and day of week also play roles. In Fig. 4, we plot the feature importance for the decision tree shown in Fig. 3.

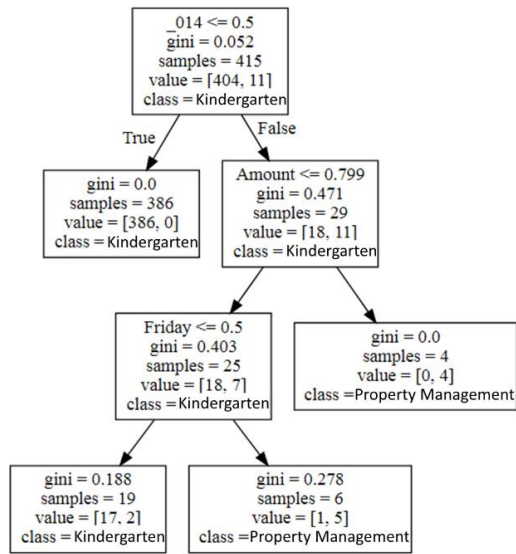


Fig. 3 The decision tree for supercluster 10 (kindergarten) makes the final distinction between transactions relating to kindergarten and those relating to property management. This is one of many trees, each relating to a single supercluster and separating instances observed in that supercluster during training.

Fig. 4 shows that the transaction code is the most important feature. This correlates well to our previous estimated of the overall and average feature importance in the original model (Fig. 1). The SHAP feature importance coincides with the feature importance observed in Fig. 4.

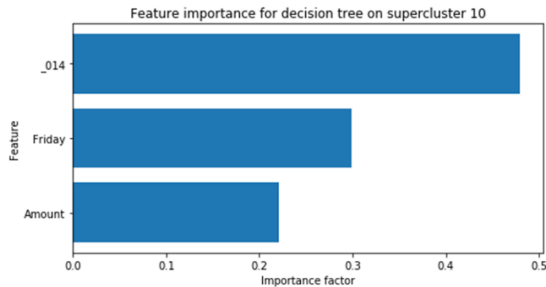


Fig. 4 Each decision tree supplies a feature importance estimate. The feature importance is shown for the decision tree of supercluster 10 (kindergarten).

The additive SHAP values allowed us to identify the transaction text as the dominant feature for transaction classification. Among the remaining features, the shallow decision trees identified the transaction code as the feature that filters transaction from heterogeneous clusters of the word2vec text embedding. In the large decision tree, the words from the transaction text were scattered throughout the nodes. It remains to be seen whether this is a general property.

We evaluated the fidelity of the explanations by comparing their prediction with those of the transaction model [20]. The explanation model made the same prediction as the transaction model for 98.3% of the labelled data (95% confidence interval of 0.1%).

To evaluate the transaction model’s robustness to changes in the transaction text, we scored a perturbed dataset and found that the model prediction typically changed for 80% of transactions with a single word replaced. We then repeated the experiment with a new set of perturbed transactions, where we replaced more than one word; this typically resulted in 90% of the transactions being classified differently.

## VII. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

In this paper, we introduced a transaction classification model which is the basis for future value adding products for banking customers, with the end goal of developing a digital financial advisor. It is thus imperative that the transaction classifier be implemented in accordance with two of the principles of responsible AI: explainability and robustness.

We found that decision trees and random forests derived from the transaction model may offer *explainability*, but their complexity (> 50 000 nodes) limits their *interpretability*.

We mitigated the complexity of the feature space by identifying the transaction text as salient. The text was then used to cluster the dataset, before fitting a small tree to each cluster where necessary. These decision trees offered improved *interpretability* as they were smaller and easier for a human to understand.

Finally, we briefly investigated the robustness of the model by subjecting it to an evasion attack. The large influence observed for text perturbations correlates well with our SHAP analysis which suggests a large model dependence on the text. We find that the model is vulnerable to changes in the transaction text. However, since vendors seldomly change their formulas for generating transaction texts and companies seldomly change their names, the text is mostly an immutable property of the transactions. This vulnerability is therefore deemed low risk for such transactions. In the case of bank transfer transactions where customers may enter free text, there could be risk of masking fraudulent or money laundering transactions. If the classifier was ever to be used to detect such transactions this would be a point to address.

## ACKNOWLEDGMENT

We would like to thank the SpareBank 1 Alliance for useful discussions and SpareBank 1 SR-Bank for providing anonymized transaction data.

## REFERENCES

- [1] J. van der Burgt, “General principles for the use of Artificial Intelligence in the financial sector,” De Nederlandsche Bank, 2019.
- [2] X. L. Zheng, M. Y. Zhu, Q. B. Li and C. C. Chen, “FinBrain: When Finance Meets AI 2.0,” *Frontiers of Information Technology & Electronic Engineering*, 2010.
- [3] M. Stefanel and U. Goyal, “Artificial Intelligence & Financial Services - Cutting through the noise,” APIS Partners, 2020.
- [4] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila and F. Herrera, “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI,” *Information Fusion*, vol. 58, no. 1, pp. 82-115, 2020.
- [5] J. Waalen and J. Olsen, “Chasing cashless? The Rise of Mobile Wallets in the Nordics,” Deloitte AS, Oslo, Norway, 2019.
- [6] McKinsey & Company, “Cash - An inefficient and Outdated Means of Payment,” McKinsey & Company, Oslo, Norway, 2019.

- [7] S. Barocas and A. Selbst, "Big Data's Desperate Impact," 104 *California Law Review* 671, 2016.
- [8] J. M. Fromholz, "The European Union Data Privacy Directive," *Berkeley Technology Law Journal*, vol. 15, no. 1, pp. 461-484, 2000.
- [9] S. Wachter, B. Mittelstadt and C. Russel, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard Journal of Law & Technology*, 2018.
- [10] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765-4774, 2017.
- [11] K. El Mokhtari, B. P. Higdon and A. Basar, "Interpreting financial time series with SHAP values," in *IBM Centers for Advanced Studies Conference (CASCON)*, Toronto, Canada, 2019.
- [12] O. Parisot, Y. Didry, B. Pierrick and B. Otjacques, "Data visualization using decision trees and clustering," in *International Conference on Information Visualization Theory and Applications (IVAPP)*, Lisbon, Portugal, 2014.
- [13] I. Polaka and A. Borisov, "Clustering-based decision tree classifier," *Technological and Economic Development of Economy*, vol. 16, no. 4, pp. 765-781, 2010.
- [14] L. Antwarg, R. M. Miller, B. Shapira and L. Rokach, "Explaining Anomalies Detected by Autoencoders Using SHAP," *Journal of Artificial Intelligence*, 2020.
- [15] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307-317, 1953.
- [16] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 22, pp. 1135-1144, 2016.
- [17] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *AAAI*, pp. 226-231, 1996.
- [18] E. Schubert, J. Sander, M. Ester, H. P. Kriegel and X. Xu, "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 19:1-19:21, 2017.
- [19] G. R. Venkata and P. A. Bhanu, "Space and Cosine Similarity measures for Text Document Clustering," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 2, 2013.
- [20] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti and D. Pedreschi, "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys*, vol. 51, no. 5, 2018.
- [21] S. Vijayalakshmi and S. Punithavalli, "A Fast Approach to Clustering Datasets using DBSCAN and Pruning Algorithms," *International Journal of Computer Applications*, vol. 60, no. 14, 2013.
- [22] M. Sato, J. Suzuki, H. Shindo and Y. Matsumoto, "Interpretable Adversarial Perturbation in Input Embedding Space for Text," *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 4323-4330, 2018.



## Appendix B

# Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality

This paper has been published as:

C. Maree and C. W. Omlin, “Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality”, *IEEE Symposium Series on Computational Intelligence (SSCI)*, **2021**, pp. 1–5, doi: 10.1109/SSCI50451.2021.9659905.

Copyright © 2021 IEEE



# Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality

Charl Maree<sup>1</sup>  
Center for AI Research  
University of Agder  
Grimstad, Norway  
charl.maree@sr-bank.no

Christian W. Omlin  
Center for AI Research  
University of Agder  
Grimstad, Norway  
christian.omlin@uia.no

**Abstract**—Customer segmentation has long been a productive field in banking. However, with new approaches to traditional problems come new opportunities. Fine-grained customer segments are notoriously elusive and one method of obtaining them is through feature extraction. It is possible to assign coefficients of standard personality traits to financial transaction classes aggregated over time. However, we have found that the clusters formed are not sufficiently discriminatory for micro-segmentation. In a novel approach, we extract temporal features with continuous values from the hidden states of neural networks predicting customers' spending personality from their financial transactions. We consider both temporal and non-sequential models, using long short-term memory (LSTM) and feed-forward neural networks, respectively. We found that recurrent neural networks produce micro-segments where feed-forward networks produce only coarse segments. Finally, we show that classification using these extracted features performs at least as well as bespoke models on two common metrics, namely loan default rate and customer liquidity index.

**Keywords**—AI in finance, feature extraction, transfer learning, recurrent neural networks, financial transactions, Big-Five personality

## I. INTRODUCTION

Effective customer engagement is critical in any retail industry and retail banking is no exception. As their customer bases grow, banks have to employ ever advancing tools to maintain if not improve the level of personalization in their interactions. AI provides such a tool and is becoming ubiquitous in retail banking [1]. In machine learning, feature extraction is the process of compressing information held in a feature set and replicating it with high fidelity using fewer features. In contrast to feature selection which selects a subset of features, feature extraction creates new features with reduced redundancy. A feature is a single quantifiable property of the data and can be numerical, categorical or textual. Dimensionality reduction is important in applications where independent data observations are finite; an increasing number of features rapidly increases the volume of the feature space such that available data quickly become sparse, counteracting the statistical significance of results. Feature extraction may also facilitate the prediction of a different but related dataset, e.g., through transfer learning which applies previously learned knowledge to a new problem. The learned relationships between the original features and the reduced features may be retained when predicting new dependent variables with fewer observations. The success of transfer learning has, for example, been demonstrated for image classification [2]. By re-using these pretrained models, smaller teams may benefit from their exceptional properties while forgoing the majority of data acquisition and preprocessing.

Transfer learning has been used in banking related applications such as customer credit scoring where knowledge was transferred across different geographical districts, and customer churn prediction where knowledge was transferred across different time periods and districts [3] [4].

Customer micro-segmentation is a promising application of AI in banking; in order to develop personalized products and services, it is important to differentiate between different types of customers [5]. Traditional customer segmentation classifies individuals along demographics such as age, gender, location, etc. so as to optimize customer interactions [6]. It produces a coarse classification which could fail to depict nuanced differences between individuals, potentially leading to discrimination e.g., in credit rating according to postal codes [7]. In contrast, micro-segmentation provides a more sophisticated classification of customers and therefore holds immense potential for personalized financial products and services. Despite the advantages, there have been no published applications of micro-segmentation of financial customers. We provide a solution through a novel approach in which we extract temporal features from the states of a recurrent neural network. We show that these features form hierarchical clusters that facilitate micro-segmentation.

We intend to develop personalized digital financial advisors that match individual customers' personalities. There is a documented correlation between financial transactions and personality [8] and evidence that spending according to personality increases happiness [9]. In this study, we extract features from the financial transactions of ca. 26,000 customers over six years. We compare the performance of feed-forward neural networks to that of recurrent neural networks in micro-segmentation; to the best of our knowledge, explicit temporal modelling of customer spending behavior has never been considered before. We show that in the state space, customer spending follows 'ski slopes', i.e., well-defined discrete trajectories with a low average change of direction. In addition, these trajectories cluster for both dominant and lesser personality traits. These trajectories are promising salient features that are novel and have the potential to be used as the basis for future personalized financial products and services.

Finally, we demonstrate the efficacy of the extracted features in a transfer learning case study predicting two common customer metrics, namely loan default rate and customer liquidity index; we show that the extracted features performed at least as well as randomly initialized models trained on larger datasets. Using these extracted features, we intend to perform a micro-segmentation of our customers to facilitate the development of personalized financial advisors.

<sup>1</sup>Strategy Innovation and Development, SpareBank 1 SR-Bank ASA, Norway.

This research was partially funded by a grant from The Norwegian Research Council; project nr 311465.

## II. RELATED WORK

Spending as evinced in financial transactions has been proven to be a promising personality predictor. In [8] the authors used a random forest to predict the Big Five personality traits – openness, conscientiousness, extraversion, agreeableness, neuroticism – from the transactions of 2,193 banking customers. They determined customer personality through the Big-Five Inventory-10 questionnaire [10]. Their reported accuracy was comparable to that of using demographics as a predictor, but they reported a higher accuracy when using more specific personality traits such as materialism and self-control. An earlier paper by the same authors also used the Big Five model and a questionnaire to determine customers’ personalities [9]. They then derived a set of coefficients – between -3 and 3 – associating 59 transaction classes with each of the Big Five personality traits. A panel of 100 evaluators rated each class’ correlation with each of the Big Five traits, from which they determined a mean correlation. An example from their study is a coefficient of -0.82 for the trait “extraversion” and the spending category “books” which suggested a mild negative association between buying books and extraversion. They used these coefficients to investigate the relationship between customer spending and their personalities and reported a causal relationship between personality-oriented spending and happiness; such spending outweighed the effect of total income. Two independent studies also used the Big Five model and found correlations between personality traits and spending [11] [12]. There clearly exists a correlation between consumer spending and personality, and the Big Five model has been a popular model for personality classification.

Generally, there are surprisingly few publications on micro-segmentation and none in the field of finance and banking. One notable publication achieved a coarse segmentation through feature extraction using customers’ Big Five personality traits along with traditional demographics and transactional data [13]. They trained both an unsupervised autoencoder and a supervised neural network with loan default probability as output. The goal was to extract features that analysts may easily visualize. They concluded that by including personality, the prediction accuracy of loan defaults improved, and they showed that they were able to cluster customers in a low dimensional space.

## III. EMPIRICAL METHODOLOGY

We extracted features from customer spending data using feed-forward and recurrent neural networks with both unsupervised (autoencoder) and supervised (predictor) architectures. We then investigated the efficacy of the extracted features in a transfer learning case study predicting loan default rate and customer liquidity index.

### A. Data

For our dataset we used the financial transactions of ca. 26,000 anonymous customers between the ages of 30 and 60 over a period of 6 years. The transactions were classified into categories, such as “groceries”, “transportation”, “savings”, etc. using the explainable AI system detailed in [14]. We then added an element of time by aggregating the transactions of each customer annually and by transaction category, normalized by annual income; each datapoint represented the annual spending distribution of each customer across the transaction categories in six time-steps. We formatted the dataset to support two types of neural networks: feed-forward

and recurrent. The dataset for the feed-forward network had the shape  $[n \times 6, m]$  where  $n = 26,000$  customers and  $m$  is the number of transaction categories. In the dataset for the recurrent network, each customer had a sequence of 6 time-steps resulting in the data shape  $[n, 6, m]$ . We split the data into training (80%) and validation (20%) sets and ran 20 experiments with randomly sampled data from the training set to determine the accuracy and confidence intervals.

### B. Spending-Evinced Personality

We used the coefficients published in [9] to calculate the Big Five personality traits from our aggregated transactions. For each customer, we calculated both annual and overall personality types across the 6-year period resulting in two datasets: a  $[n \times 6, 1]$  and a  $[n, 1]$  dataset respectively. A customer’s dominant personality trait is a delicate concept and one that is useful to introduce; we defined it as the personality trait that, of the five, had the highest absolute value. For example, a large negative extraversion score translates to a large positive introversion score; in comparisons between the traits, the absolute value must therefore be used.

### C. Feature Extraction

Feature extraction and dimensionality reduction is a mainstay in machine learning. A widely used method is principal component analysis [15]. It only identifies linear correlations between features, a shortcoming addressed by autoencoders [16]. An autoencoder is an unsupervised neural network that aims at reconstructing input data in the output layer [17]. The information is compressed by successively reducing the number of nodes in the hidden layers to reach a bottleneck. It thus learns a feature representation for a set of data with a reduced dimensionality. The underlying assumption is that these features are salient since they are able to reconstruct the information contained in the input data.

We used four neural network architectures to extract features from our classified transactions: a *feed-forward autoencoder* accepting  $[n \times 6, m]$  customer spending observations as both input and output, a *feed-forward predictor* with the same input but  $[n \times 6, 1]$  annual personality traits as output, a *recurrent autoencoder* accepting  $[n, 6, m]$  sequential spending observations as both input and output, and a *recurrent predictor* with the same input but  $[n, 1]$  personality types as output. The recurrent networks used long short-term memory (LSTM) nodes, which has been described in, e.g., [18]. The size of the networks (number of nodes and layers) were hyperparameters and optimized for each network architecture.

### D. Transfer Learning

Transfer learning – for which most of the weights of a neural network are pre-trained on a related supervised machine learning task – significantly reduces the number of samples needed in training. Knowledge may also be extracted from recurrent neural networks, as demonstrated in [19]. In this early work, the authors investigated the internal neuron activations of recurrent neural networks and managed to extract the rules that govern the model. The same authors in [20] were some of the first to demonstrate transfer learning in recurrent neural networks by initializing the network with weights learned on another dataset.

We compared the performance of the extracted features from the predictive feed-forward and recurrent neural networks to that of randomly initialized networks of identical

architectures. In this case study, we predicted two common metrics in banking: loan default rate and customer liquidity index. For the transfer learning models, we initialized the weights with those from our pretrained models, while the baseline models were randomly initialized. Pretrained weights were non-trainable. For training, we used a reduced training set of 100 randomly selected observations and ran 20 experiments to calculate the accuracy and confidence intervals. Accuracy was measured against a large validation set of ca. 5,000 observations.

#### IV. RESULTS AND DISCUSSION

##### A. Micro-Segmentation through Feature Extraction

Firstly, we found that the raw personality data – the Big Five personality scores – naturally formed fuzzy clusters along the most dominant personality trait and along that specific axis. We illustrate this phenomenon in Fig. 1 where all the points to the right of a given threshold – the vertical dotted line – represent individuals whose dominant personality trait is ‘openness’. These points naturally form a fuzzy cluster to the right of this threshold. However, we observed inconsistent customer spending patterns for shorter time windows leading to unstable clusters with customers appearing in different clusters for different time windows.

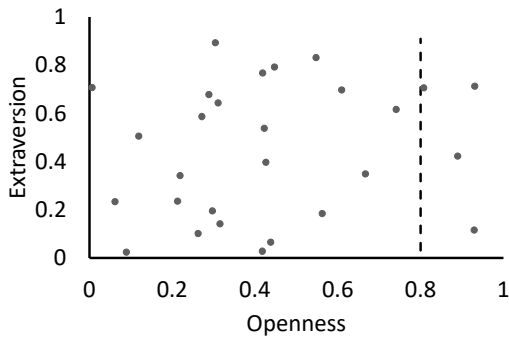


Fig. 1 An illustration of randomly distributed data naturally forming fuzzy clusters along the dimensions (axes) of the data.

Supervised feature extraction via a predictive recurrent neural network, however, yielded more constructive results. Using the established elbow-method, we determined that a network with three internal LSTM nodes was the point of diminishing returns; more than three nodes did not significantly increase the predictive accuracy, while fewer nodes substantially reduced the accuracy. These three nodes represented the extracted features and since LSTM nodes have memory the features could be visualized as trajectories in time. In Fig. 2 we visualize all combinations of the two-dimensional projections of the three-dimensional state space. We found that the extracted features for each customer followed trajectories with low average change of direction.

Furthermore, the trajectories corresponding to dominant personality traits formed clusters. We also observed a hierarchy of sub-clusters for lesser personality traits; as we zoomed into a cluster for a personality trait, we recursively found sub-clusters which corresponded to lesser personality traits. In other words, the existing hierarchy of the relative strength of the personality traits was reflected in a hierarchy of clusters of spending trajectories. This hierarchy of trajectories could be used for micro-segmentation to personalize financial recommendations. Additionally, these clusters were stable in time, as each trajectory remained in the same micro-cluster for the observed six-year time period. In this study we merely observed the presence of the clusters, but in future work we intend to apply more formal trajectory clustering techniques, which typically have a complexity of  $O(n)$ , as described in [21].

The formation of these trajectories in time is an interesting observation, since no such trajectories were present in raw input data – transaction classes aggregated in time. Interestingly, we found similar trajectories in the state space of a recurrent *autoencoder* as in the *predictor*, but with no clustering. A possible reason for the lack of trajectories with low change in direction in the raw input data is the natural inconsistency of spending; events naturally occur in people’s lives that suddenly and temporarily require a different spending pattern, e.g., large purchases such as cars or irregular expenditure such as medical bills or household repairs. However, it seems that recurrent neural networks are able to ‘smooth’ these naturally inconsistent data. We hypothesize that the recurrent neural network managed to learn temporal trajectories from the input (as observed in the autoencoder) and clustering from the output. Interestingly, features extracted from a feed-forward neural network behaved differently: though they formed clusters along the dominant personality trait, no sub-clustering was observed. Naturally, without a time element, there were also no trajectories and no temporal stabilization in feed-forward networks.

##### B. Transfer Learning Case Study

To test the efficacy of our extracted features, we benchmarked them against randomly initialized models of identical architectures. Table 1 shows the predictive performance of our extracted features on customer liquidity index, while Table 2 shows the performance when predicting loan default rate. In both cases, our extracted features performed at least as well as the randomly initialized models, but with far fewer trainable parameters. Having fewer trainable parameters has several benefits, including reduced training time and smaller dataset requirements. We also noticed that the confidence intervals were typically smaller for the transfer learning cases, suggesting improved precision.

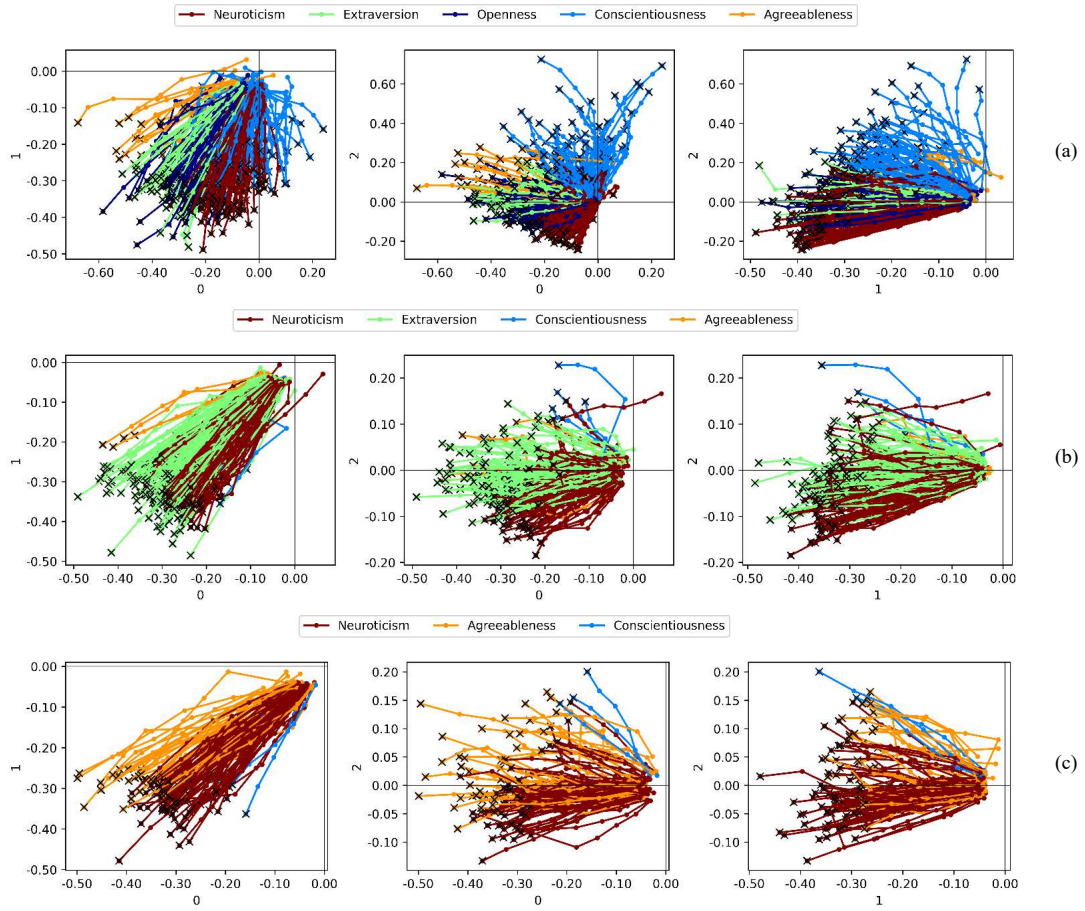


Fig. 2 Hierarchical clustering of customers' spending personalities illustrated in three parts: (a) through (c). We show the two-dimensional projections of the trajectories from the state space of a recurrent neural network where each axis represents the activation of a single node. Each trajectory represents the annual aggregated spending of a single customer for a six-year period. Part (a) shows the clustering of customers' trajectories by their dominant personality trait, while parts (b) and (c) drill down to show sub-clusters of trajectories corresponding to the second and third most dominant traits, respectively. (b) shows the sub-clusters for the parent cluster "Openness", while (c) drills down into the sub-cluster "Extraversion" from (b).

TABLE 1 A COMPARISON OF THE MEAN SQUARE ERROR VALIDATION LOSSES WHEN PREDICTING *LIQUIDITY INDEX* ON A LIMITED DATASET USING TRANSFER LEARNING VERSUS RANDOMLY INITIALIZED WEIGHTS. IN EACH CASE, THE TRANSFER LEARNING MODEL DID AT LEAST AS WELL AS A RANDOMLY INITIALIZED MODEL BUT HAD FAR FEWER TRAINABLE WEIGHTS.

Neural network Type	Weight initialization	Total weights	Trainable weights	MSE loss	95% confidence interval
Recurrent	Random	1637	1637	0.81	0.012
Recurrent	Transfer learning	1637	5	0.81	0.016
Feed-forward	Random	506	506	0.81	0.048
Feed-forward	Transfer learning	506	6	0.81	0.035

TABLE 2 A COMPARISON OF THE MEAN SQUARE ERROR VALIDATION LOSSES WHEN PREDICTING DEFAULT RATE ON A LIMITED DATASET USING TRANSFER LEARNING VERSUS RANDOMLY INITIALIZED WEIGHTS. IN EACH CASE, THE TRANSFER LEARNING MODEL DID AT LEAST AS WELL AS A RANDOMLY INITIALIZED MODEL BUT HAD FAR FEWER TRAINABLE WEIGHTS.

Neural network Type	Weight initialization	Total weights	Trainable weights	MSE loss	95% confidence interval
Recurrent	Random	1637	1637	11.4	0.048
Recurrent	Transfer learning	1637	5	11.4	0.006
Feed-forward	Random	506	506	6.7	0.125
Feed-forward	Transfer learning	506	6	6.4	0.002

## V. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

In this paper, we introduce a novel approach for customer micro-segmentation by extracting features from customers' financial transactions using recurrent neural networks. We used published coefficients to calculate customers' personalities which we used for feature extraction. We found that by using recurrent neural networks we were able to introduce an element of time to the transactions, which stabilized the extracted features and facilitated micro-segmentation. The features followed trajectories with low average changes in direction in the extracted feature space – meaning the customers remained within their micro-segments for the observed time frame – which was not the case for their spending data or their calculated personalities. These trajectories could be recursively sub-clustered according to successive dominance of customers' personality traits, leading to a hierarchy of sub-clusters. This hierarchy of customer spending trajectories is important because could be used for micro-segmentation which might facilitate personalized financial services.

We demonstrated the efficacy of our extracted features in a transfer learning case study predicting both loan default rate and customer liquidity index. We benchmarked our transfer learning models against randomly initialized models of identical architectures. We found that our extracted features performed at least as well as randomly initialized models but required far fewer trainable parameters. Fewer trainable parameters pose several benefits in a neural network, including faster training times and smaller dataset requirements.

In future work, we want to test our hypothesis that the extracted feature trajectories are robust with respect to the window of aggregation. This will be an improvement on the clustering behavior observed in spending personality, which is erratic for shorter time windows. Having such stable micro-segments will allow the development of personalized financial services, such as budgeting and savings advice. Each customer trajectory places that customer on a 'ski slope' in the state space of the recurrent neural network, indicating a pattern in spending personality. Personality is expected to play a significant role, as it has been shown that happiness is increased when spending fits personality [9]. We will also apply formal trajectory clustering methods as described in [21] and inspect the impact of noisy data and occurrence of outliers. Finally, we intend to provide a formal explanation for our extracted features and an interpretation of our model.

### ACKNOWLEDGMENT

We thank Joe Gladstone for insightful conversations about personality and spending and Perry McPartland for proofreading the first draft of the manuscript.

### REFERENCES

- [1] A. Fernández, "Artificial intelligence in financial services," The Bank of Spain, 2019.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations*, San Diego, USA, 2015.
- [3] J. Xiao, R. Wang, G. Teng and Y. Hu, "A transfer learning based classifier ensemble model for customer credit scoring," in *Seventh International Joint Conference on Computational Sciences and Optimization*, Beijing, China, 2014.
- [4] B. Zhu, J. Xiao and C. He, "A balanced transfer learning model for customer churn prediction," in *Proceedings of the Eighth International Conference on Management Science and Engineering Management*, Berlin, Germany, 2014.
- [5] E. T. Apeh, B. Gabrys and A. Schierz, "Customer profile classification using transactional data," in *Third World Congress on Nature and Biologically Inspired Computing*, Salamanca, Spain, 2011.
- [6] W. R. Smith, "Product differentiation and market segmentation as alternative marketing strategies," *The Journal of Marketing*, vol. 21, no. 1, pp. 3-8, 1956.
- [7] S. Barocas and A. Selbst, "Big data's disparate impact," *California Law Review*, vol. 104, no. 671, pp. 671-732, 2016.
- [8] J. J. Gladstone, S. C. Matz and A. Lemaire, "Can psychological traits be inferred from spending? Evidence from transaction data," *Psychological Science*, vol. 30, no. 7, pp. 1-10, 2019.
- [9] S. C. Matz, J. J. Gladstone and D. Stillwell, "Money buys happiness when spending fits our personality," *Psychological Science*, vol. 27, no. 5, pp. 715-725, 2016.
- [10] B. Rammstedt and O. P. John, "Measuring personality in one minute or less: A 10-item short version of the Big Five Inventory in English and German," *Journal of Research in Personality*, vol. 41, no. 1, pp. 203-212, 2007.
- [11] E. K. Nyhus and P. Webley, "The role of personality in household saving and borrowing behaviour," *European Journal of Personality*, vol. 15, no. 1, pp. 85-103, 2001.
- [12] L. Mangiavacchi, L. Piccoli and C. Rapallini, "Personality traits and household consumption choices (in press)," *The B.E. Journal of Economic Analysis & Policy*, 2020.
- [13] S. Mousaeirad, "Intelligent vector-based customer segmentation in the banking industry," *arXiv:2012.11876v1*, pp. 1-41, 2020.
- [14] C. Maree, J. E. Modal and C. W. Omlin, "Towards responsible AI for financial transactions," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, Australia, 2020.
- [15] J. Shlens, "A tutorial on principle component analysis," *arXiv:1404.1100v1*, pp. 1-12, 2014.
- [16] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *American Institute of Chemical Engineers (AIChE)*, vol. 37, no. 2, pp. 233-243, 1991.
- [17] S. Gu, B. Kelly and D. Xiu, "Autoencoder asset pricing models," *Journal of Econometrics*, vol. 222, no. 1, pp. 429-450, 2021.
- [18] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222-2232, 2015.
- [19] C. W. Omlin and C. L. Giles, "Extraction of rules from discrete-time recurrent neural networks," *Neural Networks*, vol. 9, no. 1, pp. 41-52, 1996.
- [20] C. W. Omlin and C. L. Giles, "Training second order recurrent neural networks using hints," in *Proceedings of the Ninth International Conference on Machine Learning*, San Mateo, USA, 1992.
- [21] J. Bian, D. Tian, Y. Tang and D. Tao, "A survey on trajectory clustering analysis," *arXiv*, vol. 1802.06971, pp. 1-40, 2018.

## Appendix C

# Understanding Spending Behavior: Recurrent Neural Network Explanation and Interpretation

This paper has been published as:

C. Maree and C. W. Omlin, “Understanding Spending Behavior: Recurrent Neural Network Explanation and Interpretation”, *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, **2022**, pp. 1–7, doi: 10.1109/CIFEr52523.2022.9776210.

Copyright © 2022 IEEE

# Understanding Spending Behavior: Recurrent Neural Network Explanation and Interpretation

Charl Maree\*

*Center for Artificial Intelligence Research  
University of Agder  
Grimstad, Norway  
charl.maree@uia.no*

Christian W. Omlin

*Center for Artificial Intelligence Research  
University of Agder  
Grimstad, Norway  
christian.omlin@uia.no*

**Abstract**—Micro-segmentation of customers in the finance sector is a nontrivial task and has been an atypical omission from recent scientific literature. Where traditional segmentation classifies customers based on coarse features such as demographics, micro-segmentation depicts more nuanced differences between individuals, bringing forth several advantages including the potential for improved personalization in financial services. AI and representation learning offer a unique opportunity to solve the problem of micro-segmentation. Although ubiquitous in many industries, the proliferation of AI in sensitive industries such as finance has become contingent on the explainability of deep models. We had previously solved the micro-segmentation problem by extracting temporal features from the state space of a recurrent neural network (RNN). However, due to the inherent opacity of RNNs, our solution lacked an explanation. In this study, we address this issue by extracting a symbolic explanation for our model and providing an interpretation of our temporal features. For the explanation, we use a linear regression model to reconstruct the features in the state space with high fidelity. We show that our linear regression coefficients have not only learned the rules used to recreate the features, but have also learned the relationships that were not directly evident in the raw data. Finally, we propose a novel method to interpret the dynamics of the state space by using the principles of inverse regression and dynamical systems to locate and label a set of attractors.

**Index Terms**—explainable AI, micro-segmentation, inverse regression, dynamical systems

## I. INTRODUCTION

Customer segmentation is an important field in banking and with customer bases growing, banks are having to employ ever advancing methods to maintain, if not improve, levels of personalization [1]. Customer segmentation has typically been achieved using demographics such as age, gender, location, etc. [2]. However, these features not only produce coarse segments, but also introduce the potential for discrimination, e.g., when using postal codes for credit rating [3]. In contrast, micro-segmentation provides a more sophisticated, fine-grained classification that depicts nuanced differences between individuals, improves personalization, and promotes

fairness. Despite these advantages and the fact that the need for such fine-grained segmentation has been highlighted [4], the scientific community has been surprisingly quiet on the topic with only a few recent publications from, e.g., the health sector [5], [6] and apparently none from the finance sector. We observe the spending behaviour of customers over time using a recurrent neural network (RNN) which allows the extraction of salient features not possible with feed-forward neural networks or otherwise [7].

Artificial intelligence is fast becoming ubiquitous across multiple industries with representation learning an auspicious method for customer micro-segmentation [7]. Sensitive industries such as finance face legal and ethical obligations towards the responsible implementation of AI [8]. The European Commission has published several guidelines surrounding responsible AI and scientific fundamentals have been consolidated in recent surveys on the topic [9], [10]. Explainability and interpretability are key elements in responsible AI [11], which are generally not yet adequately addressed in applications of AI in finance [12]. Our perspective on explainability in AI refers to a symbolic representation of a model, whereas interpretability refers to a human understanding of and reasoning about the functionality of the model. Explainability therefore neither guarantees nor implies interpretability. In this study, we address both the issues of explainability and interpretability, and we introduce a novel method for interpretation of features based on inverse regression and dynamical systems [13], [14].

Our aim is to extract and facilitate the use of salient features in future financial services; we have already shown the potential in predicting default rate and customer liquidity indices [7]. Our ultimate goal is the development of personalized financial services in which responsible customer micro-segmentation is key.

## II. RELATED WORK

### A. Representation Learning using Recurrent Neural Networks

In [15], the authors developed a model for predicting spending personality from aggregated financial transactions with the intent to investigate the causality between personality-aligned spending and happiness. They rated each of 59 spending categories according to its association with the Big-Five personality traits - extraversion, neuroticism, openness,

This work is partially funded by The Norwegian Research Foundation, project number 311465.

\*Author's second affiliation: Chief Technology Office, Sparebank 1 SR-Bank, Stavanger, Norway.

conscientiousness, and agreeableness [16] - which resulted in a set of  $59 \times 5$  linear coefficients. We used these coefficients in a previous study to train a RNN to predict customers' personality traits from their aggregated transactions [7]. In this study, we showed that the temporal features in the state space of the RNN had interesting properties: they formed smooth trajectories which formed hierarchical clusters along successive levels of dominance<sup>1</sup> of the personality traits. We also showed that similarly salient features could not be extracted from the raw data otherwise. Spending patterns over time are either more consistent than transactions aggregated over a short time period, they may fluctuate, or they may change based on life circumstances. Modelling spending over time elucidates spending patterns and thus may lead to better features [17]. Fluctuations or changes are also better represented by time series. The hierarchical clustering of the extracted features provided a means of micro-segmenting customers based on their financial behaviour. However, the responsible employment of this model demands an explanation and interpretation, which is what we address in this study.

RNNs have recently set the benchmark for human activity recognition where data from wearable sensors were used to segment and recognise activities such as gaits, steps, and gestures [18]. They are also useful to predict customer behaviour using temporal recency, frequency, and monetary data in e-commerce [19]. RNNs can be used to discriminate individuals based on their historical browsing patterns [20]. Other studies have employed RNNs to encode spatial and temporal information contained in the two-dimensional trajectories of physical objects [21], in customer churn prediction [22], [23], and to characterize individuals in recommender systems for online shopping or video streaming [24]. While RNNs are popular in such applications, few attempt to explain, interpret, and therefore understand their models. This is the contribution of our work.

### B. Explaining Recurrent Neural Networks

Finding symbolic representations of AI models is a key area of explainable AI [10]. In [25] the authors developed a symbolic regression algorithm that successfully extracted physics equations from neural networks. They managed to extract all 100 of the equations from the well known Feynman Lectures on Physics and 90% of more complicated equations, an improvement from 15% using state-of-the-art software. This was an important study because it not only proved that deep neural networks are capable of learning complicated equations and coefficients, but that it is possible to extract symbolic knowledge from such networks. The authors in [26] presented a visual method to explain RNNs used in natural language processing problems. They clustered the activations in the state space and used word clouds to visualize correlations between node activations and words in the input sentences. Similarly, the authors in [27] applied clustering

<sup>1</sup>The dominant personality trait is the one with the largest coefficient in the Big-Five model of personality traits [7].

in the state space of RNNs, but here the authors showed that *symbolic* representations could be extracted as opposed to visual explanations. Studies such as these prove that deep neural networks are indeed not inexplicable black box systems, but could be a means of discovering symbolic representations of complex relationships in data.

## III. METHODOLOGY

### A. Recurrent Neural Network Training

We used the financial transactions of approximately 26,000 customers to train a RNN to predict spending personality, as described in detail in [7]. To summarize, the input data were each customer's transactions aggregated annually across 97 transaction classes, such as groceries, transport, leisure, etc., over a period of six years. This gave an input vector  $I \in [0, 1]^{N \times T \times C}$  where  $\sum_{c=1}^C I_{n,t,c} = 1 \forall n \in [1, N], t \in [1, T]$  where  $N \simeq 26000$  customers,  $T = 6$  time-steps, and  $C = 97$  transaction classes. Each value in  $I$  therefore represents the fraction of total income spent by a given customer in a given year on a given transaction class. The output data  $O \in [-1, 1]^{N \times P}$  were the customers' Big-Five personality traits (i.e.  $P = 5$ ) calculated from published linear coefficients linking transaction classes to personality traits [15]. Our RNN consisted of three long short-term memory (LSTM) nodes [28]. The number of nodes was determined by optimizing the diminishingly increasing prediction accuracy for an increasing number of nodes, also known as the 'elbow' optimization method; RNN architectures are known to perform well with low-dimensional representations [29]. After training and during prediction, we inspected the activations of the three recurrent nodes in the state space  $S \in \mathbb{R}^{N \times T \times M}$  where  $M = 3$  is the number of LSTM nodes; each customer was represented by a trajectory with six data points in the three-dimensional space. These trajectories were our extracted features which may be used for micro-segmentation of customers [7].

### B. Explanation through Surrogate Modelling

To provide an explanation for the RNN, we trained a linear regression model - an inherently transparent class of models [10] - to replicate the trajectories from each customer's aggregated spending distribution:  $F_\theta(I) \mapsto S$  where  $\theta$  represents the coefficients of the linear regression model  $F$ . We show that these coefficients reproduced, with high fidelity, the states of the RNN, thereby offering a symbolic explanation of its functioning.

### C. Interpretation through Inverse Regression

To obtain an interpretation of the features, we propose a new method that maps the output space  $O$  onto the state space  $S$  using inverse regression [13]. From an  $M$ -dimensional grid  $S' \in \mathbb{R}^{|K| \times M}$  where  $S'_i \in \{0.1k, k \in K = [-10, 10]\}, i \in [1, M]$ , filling the entire volume of the  $M$ -dimensional state space  $S$ , and using the trained weights of the *output layer* of the RNN,  $\omega_{out} \in \mathbb{R}^{M \times (P+1)}$ <sup>2</sup>, we calculated the entire

<sup>2</sup>The dimensions  $M \times (P + 1)$  represent the weights connecting the  $M$  LSTM nodes to the  $P$  output nodes, plus one dimension to account for the bias.



reachable output as a  $P$ -dimensional hypercube  $O' \in \mathbb{R}^{|K| \times P}$ , where  $|K| = 21$  is the number of points in each dimension of the grid  $S'$ . Formally,

$$O' = S' \cdot \omega_{out}$$

This reachable hypercube of the output space is shown in Figure 5. Next, using the principles of inverse regression as described in [13], we calculated the parameters  $\omega_{inv} \in \mathbb{R}^{(P+1) \times M}$  that map the output space  $O$  to the state space  $S$ . Formally,

$$\omega_{inv} = (O'^T O')^{-1} \cdot (O'^T S')$$

In order to map the *magnitudes* of the dimensions of the output space  $O$  onto the state space  $S$ , we created a diagonal matrix  $\mathcal{D} \in \mathbb{R}^{P \times P}$  with the elements on the diagonal equal to the magnitude of each dimension of the output hypercube  $O'$ :

$$\mathcal{D} = \text{diag} \left\{ \max_{1 \leq i \leq |K|} O'_{i,j}, j \in [1..P] \right\} \quad (1)$$

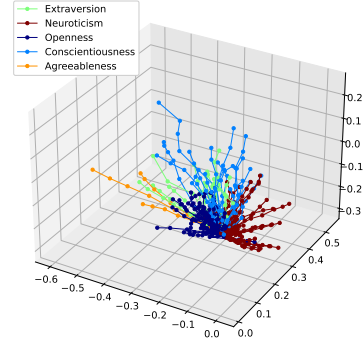
The representation of the dimensions of the output space in the state space  $\mathcal{O} \in \mathbb{R}^{P \times M}$  is then given by:

$$\mathcal{O} = \mathcal{D} \cdot \omega_{inv} - \mathbf{0}^P \cdot \omega_{inv} \quad (2)$$

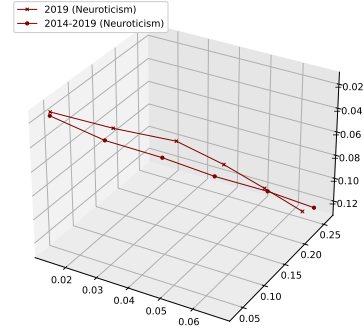
where  $\mathbf{0}^P$  is the zero vector of size  $P$  representing the origin of the output space and  $\mathbf{0}^P \cdot \omega_{inv}$  is the location of this origin in the state space.

#### IV. RESULTS

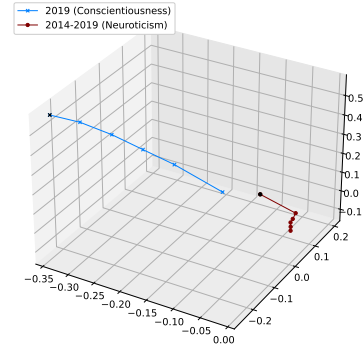
In Fig. 1, we show the features that we extracted from our RNN. Fig. 1(a) illustrates the clustering behaviour of the trajectories in the state space. Our empirical observations led us to hypothesise the existence of attractors for each of the five personality traits. Fig. 1(b) shows two trajectories for the same customer where the inputs to the RNN were aggregated over two different time periods: one year and six years. The fact that there is little difference between these two trajectories is significant; it demonstrates that the duration of the time window did not affect customer classification. This was not the case when clustering the raw personality data, where customers frequently moved between different clusters for different time periods due to variations in spending with changing life circumstances. Although we did observe significant course changes for some customers' trajectories (e.g., in Fig. 1(c)), the vast majority of customers remained in their assigned clusters for the six-year period. This stability in customer micro-segmentation is key for personalized financial services, as financial advice has to be consistent. Fig. 1(c) shows the long-term (six years) and short-term (one year) trajectories of a single customer who changed their spending behaviour such that their dominant personality type changed in the last year. In this figure it is clear that, for the final year, both trajectories moved towards the same attractor (conscientiousness), with the neuroticism attractor no longer acting upon the long-term trajectory.



(a)



(b)



(c)

Fig. 1. Trajectories in the 3-dimensional state space of a recurrent neural network trained to predict personality from aggregated transactions. While (a) shows the clustering of the trajectories of many customers according to their most dominant personality traits, (b) shows two trajectories for the same customer identically classified for two different time periods: one year vs. six years., and (c) again shows two such trajectories, but for a different customer that converged to a common attractor (conscientiousness) in the last year, after having converged to a different attractor (neuroticism) for the first five years.

To explain our model, we fit a linear regression model to reproduce the trajectories in the state space  $S$  from the RNN's input data  $I$ . From our observations in Fig. 1(b), we hypothesized that the lengths of the trajectories were not as important as their directions. We therefore simplified the trajectories and represented them by the two angles which fully describe their directions in three-dimensional space. These angles were the outputs of our linear regression model  $F_\theta(I)$ , which fit the data with a coefficient of determination of 0.78 for an unseen test set, while a more complicated polynomial regression model managed an only slightly better 0.79. Other methods such as ridge regression and decision tree regression were inferior in accuracy. Our 97 transaction classes mostly overlapped with those of the  $59 \times 5$  published coefficients and due to aggregations such as "health and fitness" being expanded to "health" and "fitness", there were  $61 \times 5$  non-zero coefficients for calculating our customers' personality traits. The linear regression model had  $69 \times 2$  non-zero<sup>3</sup> coefficients with a strong correlation with the original non-zero coefficients. Furthermore, within each of the clusters in Fig. 1(a), we observed hierarchical sub-clusters along the second, third, and fourth most dominant personality traits. This hierarchical sub-clustering is important because it provides a means of micro-segmenting customers which was not present in the raw data and could neither be replicated using feed-forward neural networks nor auto-encoders. Using our linear regression model, we created a two-dimensional plot of trajectory angles (Fig. 2). In this figure, we illustrate the hierarchical clustering behaviour that we observed for the trajectories from the RNN, where (a) shows the clustering along the customers' most dominant personality trait and (b) through (d) show the hierarchy of sub-clusters within the parent clusters. These clusters, like the trajectory clusters, were consistent in time, i.e., the linear regression model retained the desirable properties of the features from the state space of our RNN. Due to this and the high accuracy obtained in testing, we conclude that the linear regression model matched the RNN with high fidelity. The parameters  $\theta$  of the linear regression model are the symbolic explanation of the RNN, answering questions such as "Why was Customer A classified in this way?" by referring to the customer's aggregated transactions in the input data  $I$ .

We observed that the directions of the trajectories were consistent with the grades of the customers' membership in each of the five personality traits, i.e., the output data  $O$  of the RNN. The greater a customer's membership in the dominant personality trait, the quicker the trajectories converged towards the corresponding hypothesised attractor. The attractors acted not only on the dominant personality trait, but also on succeeding lesser personality traits with succeeding lesser forces. We demonstrate this in Fig. 2 where the sub-clusters preserve the structure of their parent clusters: the trajectories of lesser personality traits also converged to their respective

<sup>3</sup>Non-zero here refers to coefficients with values that are not insignificantly small compared to the mean value of all the coefficients.

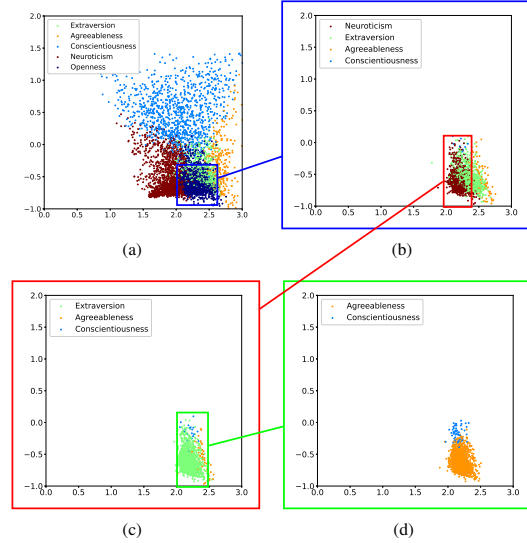


Fig. 2. Hierarchical clustering of trajectory angles in 2-dimensional space. Each axis represents an angle (in radians) which describes the direction of the trajectories in 3-dimensional space and each data point represents a trajectory. These points can be interpreted as the locations where the trajectories penetrate a sphere enclosing the state space. We show all the levels of hierarchical clustering: (a) shows the highest level, while (b) through (d) show sub-clustering within each of the subsequent parent clusters.

attractors. Intuitively, people spend differently according to their dominant personality trait. Within a group of their peers, their lesser personality traits still differentiate them from each other. Thus, the hierarchical clustering of trajectories and the labeling of the attractors is the model interpretation. Based on this observation and to locate and label the attractors, we mapped the dimensions of the output space  $O$  onto the state space  $S$  using inverse regression, as described in Section III-C. The resulting mapping ( $O$ ) is shown in Figure 6 where each colored axis represents a personality dimension. These are the axes along which customers' trajectories moved in time; each time-step moved a trajectory further along these dimensions, with the direction dictated by the grades of membership in each of the output dimensions. We proved this by predicting the final location in the state space ( $\mathcal{L}$ ) of each trajectory given the normalized grades of membership in each of the dimensions in the output space  $O$ .

$$\mathcal{L} = O^T \cdot O'^T \quad (3)$$

$$O'_j = \frac{O_j}{\max_{1 \leq i \leq |K|} O_{i,j}}, \quad j \in [1..P]$$

Figure 3 shows the predicted final locations ( $\mathcal{L}$ ) of customers' extended trajectories in the state space. We calculated these extended trajectories  $I' \in [0, 1]^{N \times T' \times C}$  by extending the number of time-steps to  $T' = 100$ , such that  $I'_{n,t',c} =$

$mean_{t \in [1, T]}(I_{n, t, c}) \quad \forall n \in [1, N], t' \in [1, T'], c \in [1, C]$ . This extension was intended to allow a larger number of time-steps such that the state space trajectories may converge to their predicted final locations  $\mathcal{L}$ . Note that though all trajectories *asymptotically* converged towards their predicted final locations, some did not fully converge. Using the extended trajectories from Fig 3, we estimated the locations of the attractors, shown in Fig 4. For three of the personality traits - agreeableness, extraversion and neuroticism - we observed line attractors which we located by fitting second-order polynomial functions to the final locations of the trajectories. For the remainder of the personality traits - openness and conscientiousness - we observed point attractors, with conscientiousness having three separate point attractors. We located these attractors by taking the means of the clusters as determined by their dominant personality traits. Since the locations of the attractors corresponded to the predicted final locations for the trajectories  $\mathcal{L}$ , we could use these locations to label the attractors according to the  $P$  personality dimensions in the output space  $O$ . The interpretation of the state space dynamics is therefore the locations and labels of the attractors based on customers' personality traits.

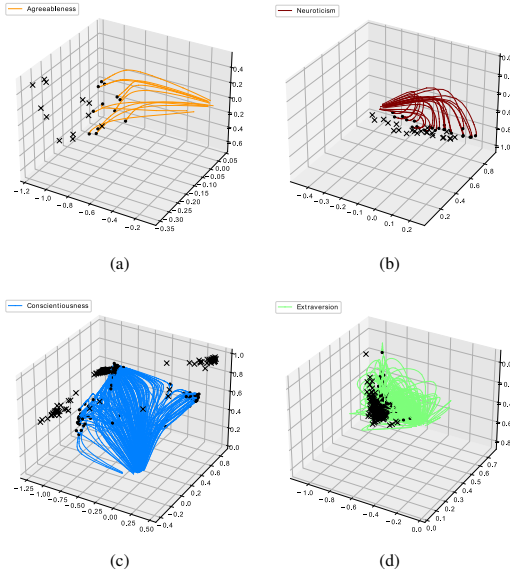


Fig. 3. Extended customer trajectories ( $I'$ ) asymptotically converging to their predicted final locations ( $\mathcal{L}$ ) in the state space, shown as  $X$ 's. Each of the sub-figures show a different cluster of customer trajectories, each having a different dominant personality trait.

## V. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

The financial sector is experiencing an increased demand in the level of personalization offered to its customers, which requires more nuanced segmentation techniques than the current offerings from traditional features such as demographics.

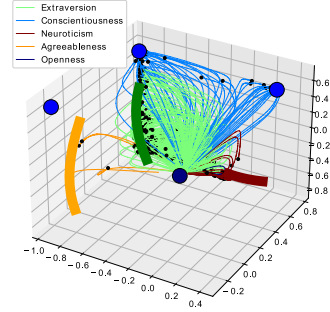


Fig. 4. A subset of trajectories in  $I'$  converging to their relevant attractors as determined by their dominant personality traits. The attractors are colored according to their corresponding personality traits and shown as polynomial lines (for line attractors) and circles (for point attractors). For readability, these attractors are drawn oversized as thick lines or circles.

Representation learning offers such an alternative technique for fine-grained segmentation, but it is plagued by the inherent opacity introduced by deep learning; explainability and interpretability promote understanding and are key in sensitive industries such as finance which must comply with regulations regarding the responsible use of AI. We proposed a solution for micro-segmentation of customers by extracting temporal features from the state space of a RNN, which formed clusters of trajectories along the most dominant of the Big-Five personality traits. Within each such cluster, we found a hierarchy of sub-clusters which corresponded to the successive levels of dominance of the personality traits. While the clusters of trajectories corresponding to the dominant personalities provide a coarse customer segmentation, the hierarchy of trajectory clusters associated with lesser personality traits offers the opportunity for micro-segmentation.

In this study, we provided a symbolic *explanation* for the RNN through a high fidelity linear regression model which answers questions such as “Why was Customer A classified in this way?” by referring to their historic financial transactions. Further, we provided an *interpretation* of the feature trajectories by applying inverse regression to map the personality dimensions into the state space, which allowed us to locate and label the attractors that govern the dynamics of the state space.

In future work, we intend to use our explainable features in the development of personal financial services such as personalized savings advice, advanced product recommendations, and wealth forecasters. There also exists the potential for a formal exploration of the attractor space through dynamical analyses to both qualify and quantify the nature of the attractors; the null space could potentially be used in a singular value decomposition to determine the major contributing inputs, as an alternative to SHAP [30].

## ACKNOWLEDGMENTS

We are grateful for fruitful discussions with Joe Gladstone on the topic of personality traits and the determination of their corresponding coefficients and with Peter Tino, Andrea Ceni, and Peter Ashwin on the topic of dynamical systems and how they apply to the evaluation of state spaces of RNNs.

## REFERENCES

- [1] M. Stefanel and U. Goyal, "Artificial intelligence & financial services: Cutting through the noise," APIS partners, London, England, Tech. Rep., 2019.
- [2] P. Kalia, "Product category vs demographics: Comparison of past and future purchase intentions of e-shoppers," *International Journal of E-Adoption (IJE)*, vol. 10, no. 2, pp. 20–37, 2018.
- [3] S. Barocas and A. D. Selbst, "Big data's disparate impact," *California Law Review*, vol. 104, no. 671, pp. 671–732, 2016.
- [4] R. Krishnapuram and A. Mondal, "Upcoming research challenges in the financial services industry: a technology perspective," *IDRBT Journal of Banking Technology*, vol. 1, no. 1, pp. 66–84, 2017.
- [5] K. Kuwayama, H. Miyaguchi, Y. T. Iwata, T. Kanamori, K. Tsujikawa, T. Yamamuro, H. Segawa, and H. Inoue, "Strong evidence of drug-facilitated crimes by hair analysis using lc–ms/ms after micro-segmentation," *Forensic Toxicology*, vol. 37, no. 1, pp. 480–487, 2019.
- [6] E. Nandapala, K. Jayasena, and R. Rathnayaka, "Behavior segmentation based micro-segmentation approach for health insurance industry," *2nd International Conference on Advancements in Computing (ICAC)*, vol. 1, no. 1, pp. 333–338, 2020.
- [7] C. Maree and C. W. Omlin, "Clustering in recurrent neural networks for micro-segmentation using spending personality," *IEEE Symposium Series on Computational Intelligence*, 2021.
- [8] J. van der Burgt, "General principles for the use of artificial intelligence in the financial sector," De Nederlandsche Bank, Amsterdam, The Netherlands, Tech. Rep., 2019.
- [9] European-Commission, "On artificial intelligence - a european approach to excellence and trust (whitepaper)," European Commission, Brussels, Belgium, Tech. Rep., 2020.
- [10] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, no. 1, pp. 82–115, 2020.
- [11] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a right to explanation," *AI Magazine*, vol. 38, no. 3, pp. 50–57, 2017.
- [12] L. Cao, "Ai in finance: Challenges, techniques and opportunities," *Banking & Insurance eJournal*, 2021.
- [13] P. A. Parker, G. G. Vining, S. R. Wilson, J. L. Szarka III, and N. G. Johnson, "The prediction properties of classical and inverse regression for the simple linear calibration problem," *Journal of Quality Technology*, vol. 42, no. 4, pp. 1–16, 2010.
- [14] A. Ceni, P. Ashwin, and L. Livi, "Interpreting recurrent neural networks behaviour via excitable network attractors," *Cognitive Computation*, vol. 12, no. 2, pp. 330–356, 2019.
- [15] S. Matz, J. Gladstone, and D. Stillwell, "Money buys happiness when spending fits our personality," *Psychological science*, vol. 27, 04 2016.
- [16] B. De Raad, "The big five personality factors: The psycholexical approach to personality," *Hogrefe & Huber Publishers*, 2000.
- [17] Y. Zhang, T. Zhou, X. Huang, L. Cao, and Q. Zhou, "Fault diagnosis of rotating machinery based on recurrent neural networks," *Measurement*, vol. 171, p. 108774, 2021.
- [18] C. F. Martindale, V. Christlein, P. Klumpp, and B. M. Eskofier, "Wearables-based multi-task gait and activity segmentation using recurrent neural networks," *Neurocomputing*, vol. 432, pp. 250–261, 2021.
- [19] H. Salehinejad and S. Rahnamayan, "Customer shopping pattern prediction: A recurrent neural network approach," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–6.
- [20] S. Vamosi, T. Reutterer, and M. Platzer, "A deep recurrent neural network approach to learn sequence similarities for user-identification," *Decision Support Systems*, p. 113718, 2022.

- [21] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," *International Joint Conference on Neural Networks (IJCNN)*, pp. 3880–3887, 2017.
- [22] C. G. Mena, A. D. Caigny, K. Coussement, K. W. D. Bock, and S. Lessmann, "Churn prediction with sequential data and deep neural networks. a comparative analysis," *ArXiv*, vol. abs/1909.11114, 2019.
- [23] J. Hu, Y. Zhuang, J. Yang, L. Lei, M. Huang, R. Zhu, and S. Dong, "pRNN: A recurrent neural network based approach for customer churn prediction in telecommunication sector," in *2018 IEEE International Conference on Big Data (Big Data)*, 2018, pp. 4081–4085.
- [24] S. Li and H. Zhao, "A survey on representation learning for user modeling," *International Joint Conferences on Artificial Intelligence Organization (IJCAI)*, pp. 4997–5003, 2020.
- [25] S.-M. Udrescu and M. Tegmark, "Ai feynman: a physics-inspired method for symbolic regression," *arXiv*, vol. 1905.11481, 2020.
- [26] Y. Ming, S. Cao, R. Zhang, Z. Li, Y. Chen, Y. Song, and H. Qu, "Understanding hidden memories of recurrent neural networks," *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 13–24, 2017.
- [27] C. W. Omlin and L. Giles, "Extraction of rules from discrete-time recurrent neural networks," *Neural Networks*, vol. 9, no. 1, pp. 41–53, 1996.
- [28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [29] N. Maheswaranathan, A. H. Williams, M. D. Golub, S. Ganguli, and D. Sussillo, "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics," *Advances in neural information processing systems (NIPS)*, vol. 32, pp. 15 696–15 705, 2019.
- [30] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4765–4774.

## APPENDIX

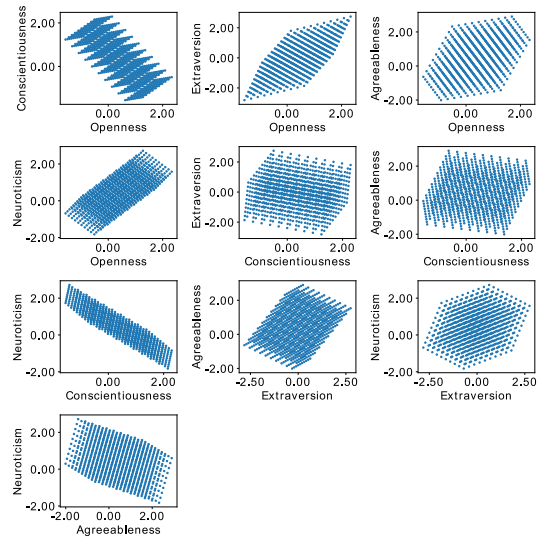


Fig. 5. The reachable output space of our RNN shown as two-dimensional projections of all combinations of the five output dimensions. The reachable output space was mapped from the reachable region in state space ( $S^5 \in [-1..1]^5$ ) using the output weights of the RNN

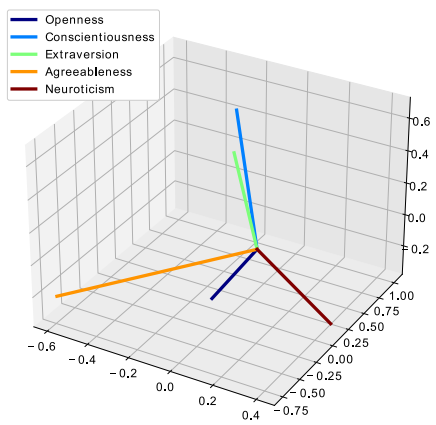


Fig. 6. The dimensions of the output space of our RNN ( $O$ ) mapped onto the state space ( $S$ ) as per Equation 2. Each coloured line represents a different labelled dimension in  $\mathcal{O}$ , with the lengths of the lines mapped from the maximum observed values of their corresponding output dimensions (Equation 1).

# Appendix D

## Balancing Profit, Risk, and Sustainability for Portfolio Management

This paper has been published as:

C. Maree and C. W. Omlin, “Balancing Profit, Risk, and Sustainability for Portfolio Management”, *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, **2022**, pp. 1–8, doi: 10.1109/CIFEr52523.2022.9776048.

Copyright © 2022 IEEE

# Balancing Profit, Risk, and Sustainability for Portfolio Management

Charl Maree\*  
Center for AI Research  
University of Agder  
Grimstad, Norway  
charl.maree@uia.no

Christian W. Omlin  
Center for AI Research  
University of Agder  
Grimstad, Norway  
christian.omlin@uia.no

**Abstract**—Stock portfolio optimization is the process of continuous reallocation of funds to a selection of stocks. This is a particularly well-suited problem for reinforcement learning, as daily rewards are compounding and objective functions may include more than just profit, e.g., risk and sustainability. We developed a novel utility function with the Sharpe ratio representing risk and the environmental, social, and governance score (ESG) representing sustainability. We show that a state-of-the-art policy gradient method – multi-agent deep deterministic policy gradients (MADDPG) – fails to find the optimum policy due to flat policy gradients and we therefore replaced gradient descent with a genetic algorithm for parameter optimization. We show that our system outperforms MADDPG while improving on deep Q-learning approaches by allowing for continuous action spaces. Crucially, by incorporating risk and sustainability criteria in the utility function, we improve on the state-of-the-art in reinforcement learning for portfolio optimization; risk and sustainability are essential in any modern trading strategy, and we propose a system that does not merely report these metrics, but that actively optimizes the portfolio to improve on them.

**Keywords**—AI in finance, multi-agent reinforcement learning, genetic algorithms, MADDPG

## I. INTRODUCTION

Stock portfolio optimization has been a focal point in financial technology with various solutions proposed including artificial neural networks, support vector machines, random forests, and, more recently, reinforcement learning [1, 2]. The application of reinforcement learning to stock portfolio optimization has generally followed two different approaches: deep Q-learning (DQL) where discretized actions denote buy and sell volumes [3], and policy gradient methods where continuous actions correspond to the distribution of assets in the portfolio [4]. In recent publications, DQL has typically been outperforming policy gradient methods even though discretization is considered disadvantageous [5]. We therefore investigate the cause of the inferior performance of policy gradient methods and propose a solution: replacing gradient descent with a genetic algorithm for parameter optimization. Further, we note that recent studies have typically been using financial returns as the sole performance metric [6]. We propose to include two additional metrics – risk and sustainability – in a novel utility function using the Sharpe ratio and environmental, social, and governance (ESG) score, respectively. While risk is a key element of modern portfolio theory, sustainability is increasingly becoming requisite in financial services. By adding these two metrics to the utility function, we create a system that actively reduces risk while maintaining a sustainable portfolio, thus furthering the state-of-the-art in modern portfolio management.

## II. BACKGROUND AND RELATED WORK

### A. Portfolio Metrics and Market Indicators

The Sharpe ratio is commonly used to quantify the risk-to-reward ratio of a portfolio [7]. It is defined as the expected return in excess of the risk-free return per unit of risk in the portfolio, formally:

$$\text{Sharpe ratio} = \frac{R_p - R_f}{\sigma_p} \quad (1)$$

Here,  $R_p$  and  $R_f$  are the expected daily return of the portfolio and the risk-free return respectively, while  $\sigma_p$  is the standard deviation of the daily returns of the portfolio. The higher the Sharpe ratio of a portfolio, the better the risk-adjusted performance: a Sharpe ratio less than one is considered sub-optimal by investors, while a ratio greater than one is considered good, greater than two is very good, and greater than three is excellent [8]. The use of the Sharpe ratio in the reward function can significantly increase the return [9].

The environmental, social, and governance (ESG) score is a set of criteria that measure a company's operations for sustainability. It is used by socially aware investors and investment firms to select stocks appropriate to their portfolio, as well as in the finance sector generally; firms such as JPMorgan Chase, Wells Fargo, and Goldman Sachs have all published annual reports that present their ESG performances [10, 11, 12]. While the main purpose of ESG is to provide a measure of sustainable conduct, it may also serve as an indicator of long-term risk; through prioritizing ESG, an investor might be able to avoid companies that conduct high-risk activities with potential future consequences on stock prices. In this study, we use the ESG score reported by Yahoo Finance with a scale of 0-100, where a lower score indicates more sustainable conduct.

Momentum indicators are popular tools used by investors to gauge the strength of a stock. They evaluate the ability of a stock to sustain a rate of price change. Moving average convergence divergence (MACD) is one such indicator which subtracts the 26-day from the 12-day exponential moving average (EMA) – an exponentially weighted moving average, assigning more weight to recent data – of a stock price. MACD is used to predict reversals in trends but is prone to false positives, i.e., it occasionally predicts reversals that do not actually occur. The relative strength index (RSI) is another momentum indicator which is often used in tandem with MACD to mitigate this shortcoming. It uses the magnitude of recent price changes to predict overbought and oversold conditions of a given stock. RSI is calculated as follows:

\*Second Affiliation: Chief Technology Office, SpareBank 1 SR-Bank, Stavanger, Norway.

$$RSI = 100 - \frac{100}{1 + \frac{P_x}{N_x}} \quad (2)$$

Here,  $P_x$  and  $N_x$  are the averages of the positive and negative close prices respectively, for a period of  $x$  days. The RSI value lies between 0 and 100, and the typical interpretation is that values below 30 and above 70 indicate the stock being oversold and overbought, respectively. Studies have shown that MACD and RSI can increase returns for stock trading. [13, 14].

The final indicator we used is drawdown, specifically daily drawdown (DDD) and maximum drawdown (MDD). While the former is calculated as the scaled difference between the current ( $P_{current}$ ) and maximum ( $P_{max}$ ) stock prices for a given period, the latter is the scaled difference between the minimum ( $P_{min}$ ) and maximum ( $P_{max}$ ):

$$DDD = \frac{P_{current} - P_{max}}{P_{max}} \quad (3)$$

$$MDD = \frac{P_{min} - P_{max}}{P_{max}} \quad (4)$$

Drawdown is one of the most widely used indicators of risk and is a measure of *downside* volatility, in contrast to the Sharpe ratio which is a measure of volatility in general [15]. It is therefore especially useful to, e.g., short-term investors to whom *upside* volatility is not of paramount concern.

#### B. Non-Stationarity in Reinforcement Learning

In reinforcement learning, agents learn policies by maximizing expected cumulative rewards [16]; the value of each state in a Markov decision process (MDP) is the discounted sum of rewards of future states, formalized by the Bellman equation [17]:

$$V(s) = \max_{a \in A(s)} \sum_{s'} P(s' | s, a) (R(s, a, s') + \gamma V(s')) \quad (5)$$

Here,  $V(s)$  is the value of state  $s$ ,  $P(s' | s, a)$  is the probability of transitioning to state  $s'$  given state  $s$  and action  $a$ ,  $R(s, a, s')$  is the reward for action  $a$  in state  $s$  transitioning to state  $s'$ , and  $\gamma \in [0, 1]$  is the discount rate which reduces the weight of future rewards. The value of a state is the maximum discounted reward for all possible actions for that state,  $A(s)$ . While Equation (5) is the general Bellman equation for stochastic MDPs, deterministic MDPs will have the transition probability distribution  $P(s' | s, a)$  reduced to one. Furthermore, stochastic systems may either be stationary or non-stationary. Unlike stationary systems which have constant transition probability distributions, non-stationary systems have proven problematic for traditional reinforcement learning methods [18]. A relevant example of a non-stationary MDP is a multi-agent system where multiple independent agents act on the same environment resulting in unstable state transition probabilities caused by the changing policies of the other agents during training [18].

#### C. MADDPG for Stabilizing a Multi-Agent System

Multi-agent deep deterministic policy gradient (MADDPG) was introduced to address the inherent non-

stationarity of multi-agent systems [18]. In their paper, the authors demonstrated an increasing variance in policy gradients with an increasing number of agents. They extended deep deterministic policy gradient (DDPG) in which the parameters  $\theta$  of the optimum policy  $\pi^*$  are determined through maximizing the objective function  $J(\theta) = E_{s \sim p^\pi, a \sim \pi(\theta)}[R]$ , where  $p^\pi$  is the state distribution and  $\pi(\theta)$  is the policy according to parameters  $\theta$ . They formalized the gradient of the objective function for deterministic policies ( $\mu_\theta: S \mapsto A$ ) as:

$$\nabla_\theta J(\theta) = E_s [\nabla_\theta \mu_\theta(a | s) \nabla_a Q^\mu(s, a) |_{a=\mu_\theta(s)}] \quad (6)$$

In DDPG,  $\mu_\theta(a | s)$  is modelled by an actor network which predicts the best action given a state, while the reward function  $Q^\mu(s, a)$  is modelled by a critic network which estimates the value of a state-action pair. These networks experience high variance in their policy gradients when used in multi-agent settings, as the actions of other agents are absent in the loss function while the rewards depend on these actions [18]. The authors in [18] mitigated this problem by extending the critic  $Q^\mu(s, a)$  to consider the actions of all agents:  $Q^\mu(s, a_i, i \in \{1 \dots N\})$ , where  $N$  is the number of agents.

#### D. Genetic Algorithms for Parameter Optimization

In general, genetic algorithms (GA) solve problems by evolving a population of individuals  $a_i, i \in \{1 \dots N\}$ , each with a set of parameters  $\theta_i$ . At each generation  $g$ , a fitness score  $F(a | \theta_{i,g})$  is calculated for each individual through measuring their performance at solving a given problem. Typically, the best performing individual is carried over to the next generation ( $g + 1$ ), while the top  $k < N$  individuals are used to generate a new batch of  $N - 1$  individuals, such that the size of the population remains constant. This new population is generated either through parameter mutation – where parameters are altered through crossover-mutation between parents' parameters – or through the addition of Gaussian noise:  $\theta_{g+1} = \theta_g + \sigma \epsilon$  where  $\epsilon \sim \mathcal{N}(0, I)$  and  $\sigma$  is a hyperparameter which roughly corresponds to a learning rate. In [19], the authors used the addition of Gaussian noise to evolve the parameters of a neural network and found that it outperformed both DQL and gradient-based methods at playing games<sup>1</sup>. In another study, the authors used GA to evolve the parameters of a single agent system and showed that it outperformed DDPG in moving a physical robotic arm [20].

#### E. Reinforcement Learning for Stock Portfolio Optimization

In stock portfolio optimization, a trader continuously redistributes funds between a selection of stocks. Risk-aware traders structure their portfolios to optimize risk for a given expected return [21]; one approach is portfolio optimization using reinforcement learning. In [22], the authors compared the performance of different single-agent policy gradient methods on an MDP structured as follows:

- *State*: the close-price history, high-price history and a wavelet transform of the close-price for each of six stocks for a given time window.

<sup>1</sup> Games are a popular application for reinforcement learning as they facilitate learning on high-dimensional input data akin to human sensory input such as vision [29].



- *Action*: a continuous daily distribution of funds across the six stocks.
- *Reward*:  $\log(\Delta P) + S$ , where  $\Delta P$  is the daily change in portfolio value, and  $S$  is the Sharpe value.

It could be argued that this approach does not appropriately weight risk for all types of investors; certain investors might be more risk-averse than others, e.g., individuals in different stages of their lives. The authors stated that even though DDPG was their best performing method, it performed rather poorly and frequently ended in local minima. Their best performing scenario with a careful stock selection achieved approximately 25% annual returns.

Similarly, the authors in [23] presented a DDPG-based method for trading a selection of 8 stocks. They used LSTM networks for the critics and feed-forward networks for the actors. Their state consisted of daily stock prices, RSI, stock positions, and the portfolio value. Their rewards were simple daily returns, and their actions were the continuous distribution of stock positions in the portfolio. They reported compound annual return of 14% and a Sharpe ratio of 0.6 over a period of 11 years.

In [24], the authors presented a multi-agent DQL system that traded four different crypto currencies – Bitcoin (BTC), Litecoin (LTC), Ethereum (ETH), and Ripple (XRP). In this system, each agent traded a single asset and the MDP was formalized as follows:

- *State*: the close price for each asset at the given time step.
- *Action*:  $2 \times 30$  discretized bins for buy and sell respectively, and one action to hold, totaling 61 actions.
- *Reward*: two reward functions were tested: a simple sum of financial returns and a weighted sum of the returns and the Sharpe ratio.

The weighting between the returns and Sharpe ratio was a hyperparameter – an improvement over [22] as this potentially allows for different strategies depending on the investor’s appetite for risk. The authors reported that the second reward function yielded better results. They reported daily returns between 2.0% and 4.7%, while the best annualized Sharpe ratio achieved was 3.2. This system clearly performed better than the ones in [22] and [23], which could be related to the nature of the optimizer in a discretized action space; DQL does not rely on policy gradients and is therefore not susceptible to local minima. Another difference is that this study used multiple agents, i.e., one agent per stock. It could be argued, however, that these agents were simply clones that fulfilled the same role given the same observations and rewards, and that they could learn neither unique behaviors nor cooperation.

The DQL system presented in [25] divided the portfolio optimization problem into timing and pricing elements which resulted in two types of agents: signal agents and order agents, respectively. Additionally, each of these two types of agents were concerned with either buying or selling of assets, which resulted in four individual agents. Agents had individual *state* observations: while buy and sell signal agents received a history of asset prices, the sell agent also received information about potential profit using the next-day stock price. Further,

the buy and sell order agents’ observations were market indicators – the Granville indicator<sup>2</sup> and Japanese Candlesticks<sup>3</sup>. The *action* spaces for the four agents consisted of buy and sell signals sent from the signal agents to the appropriate order agents which in turn generated discrete buy or sell volumes. The *reward* function  $R \in [0,1]$  was the normalized difference between the selling or buying price and the high or low price of the next day, respectively. Though the authors presented their results in percentage profit over a 4.5-year test period (1138.7%), we calculated their compound annual return for their best-case scenario as 74.9%. They did not report a Sharpe ratio for their optimized portfolio.

In summary, discretized DQL systems typically outperform policy gradient systems for stock portfolio optimization. We hypothesized that this could be attributed to the nature of the policy gradients; flat policy gradients and local minima pose challenges for gradient-based optimizers [18]. We therefore replaced gradient descent with a genetic algorithm for parameter optimization to eliminate gradient-based optimization problems while maintaining a continuous action space. A continuous action space is desirable because stock trading is not inherently discrete, and discretization adds an unnecessary level of abstraction [5]. Finally, there has not – to the best of our knowledge – been any published reinforcement learning portfolio optimization system that used ESG in its utility function. This is a significant oversight since sustainable investing is pivotal to a more sustainable society [10]. We therefore address this by incorporating ESG in our utility function.

### III. EMPIRICAL METHODOLOGY

#### A. Data

We used market data as reported by Yahoo Finance for a selection three of stocks from the DOW30 index: The Goldman Sachs Group, Inc. (GS), The Procter & Gamble Company (PG), and 3M Company (MMM). For our training and testing periods, these three stocks had constant ESG risk scores of 28.12, 25.10, and 34.88, respectively. However, it is possible that ESG scores can change in time according to changes in companies’ operations and our system is designed to cope with such changes. We used the asset close prices for a period of two years in training (shown in Fig. 1a) and the following year in testing (shown in Fig. 1b); it is injudicious in stock portfolio optimization to not have a separate test set, firstly because trading will always happen on unseen data, and secondly because typical MDPs for stock portfolio optimization are non-stationary and therefore render reinforcement learning agents susceptible to overfitting [26, 27].

#### B. Design of Markov Decision Process

Many studies have been avoiding policy gradient methods for portfolio optimization by discretizing action spaces. However, the portfolio optimization problem is not inherently discrete and continuous action spaces are therefore considered preferable [5]. In this study, we used a triple agent system and the following MDP with a continuous action space:

<sup>2</sup> The Granville indicator is a set of eight conditions of a stock price in relation to its moving average, e.g., a bullish breakthrough is when the stock price crosses the moving average in an upward trend. It indicates buying or selling conditions.

<sup>3</sup> Japanese candlesticks consider four daily price points: open, close, high, and low. A stock is considered either bearish or bullish depending on the difference between open and close prices while the high and low prices indicate daily volatility.

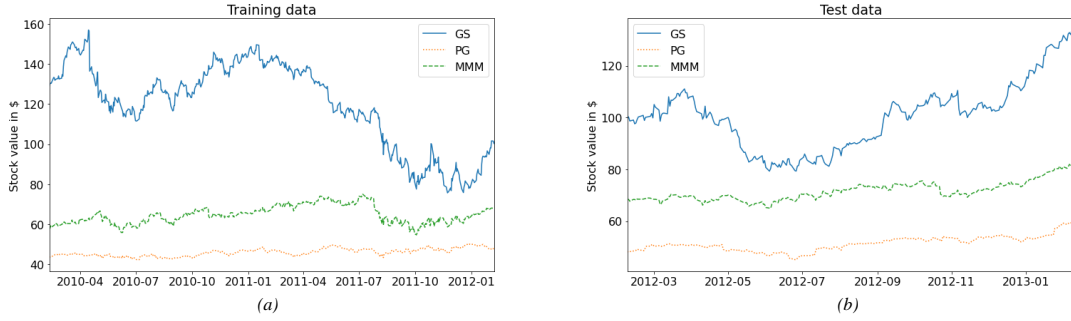


Fig. 1 The stock price data used for (a) training and (b) testing purposes. Market data is shown for The Goldman Sachs Group, Inc. (GS), The Procter & Gamble Company (PG), and 3M Company (MMM). Market conditions were slightly different between these two datasets, e.g., GS experienced an overall decrease in stock price for the training period, but an overall increase during the testing period.

- Our *state* was represented by 18 values: for each of the three stocks a normalized stock price (with subtracted mean, scaled to unit variance), MACD, RSI, DDD, MDD, and the difference between the 20-day and 5-day EMA's.
- The *action*-spaces of our first two agents (profit agent and risk-averse agent) were the continuous distributions of positions for the three stocks and one for holding cash, i.e., there were four values per action ( $A_1$  and  $A_2$  respectively where  $|A_i| = 4$ ,  $A_{i,j} \in [0,1]$ ,  $\sum_{j=1}^4 A_{i,j} = 1$ ,  $i \in \{1,2\}$ ). The third action (the sustainability action) consistently selected best performing stock with respect to ESG, e.g.,  $A_3 = [0,1,0,0]$  while PG had the lowest ESG score. The final agent's (manager agent) action was the weighting between the three actions (profit, risk, and sustainability):  $A_t = \sum_{i=1}^3 \beta_i A_{i,t}$ , where  $A_t$  was the total action sent to the environment at time-step  $t$  and  $\beta_i \in [0,1]$ ,  $i \in [1,3]$  was the third agent's action.
- The rewards were unique to each agent: the profit agent's reward was the change in portfolio value from time-step  $t$  to  $t + 1$ :  $r_{1,t} = \Delta P|_t^{t+1}$ ; it was only concerned with profit. The risk-averse agent's reward was the Sharpe ratio for a moving window of 20 days:  $r_{2,t} = \text{Sharpe}(t - 20 \rightarrow t)$  if  $t \geq 20$ , else 0; it was concerned with risk and the variability of daily returns. The manager agent's reward was a linearly weighted function of the rewards of the first two agents and the mean ESG score for the portfolio:  $r_4 = \sum_{i=1}^3 \omega_i r_{i,t}$ ,  $\sum_{i=1}^3 \omega_i = 1$ ,  $\omega_i \in [0,1]$  where  $r_{3,t} = -\sum_{i=1}^3 (x_i \cdot \text{ESG}_i)$  where  $x_i$  and  $\text{ESG}_i$  are the position and ESG score of stock  $i$  and the weighting parameters  $\omega_i$ ,  $i \in [1,3]$  were hyperparameters which we tuned to the values of 0.7, 0.2 and 0.1, respectively; the manager agent weighed the recommended actions from the other agents to achieve balanced rewards given a tunable prioritization between risk, reward and sustainability.

Finally, we calculated our market indicators as follows: We used standard periods of 14 days in Equation (2) for RSI and 26 days in Equations (3) and (4) for DDD and MDD. We annualized the Sharpe ratio by assuming 252 trading days per year  $\text{Sharpe}_{\text{annual}} = \sqrt{252} \cdot \text{Sharpe}_{\text{daily}}$  where  $\text{Sharpe}_{\text{daily}}$  was calculated from Equation (1) with a risk-free return equal to zero; we assumed risk-free returns were negligible which is not an unusual assumption with consistently low interest rates for our selected time period. For simplicity, we ignored transaction costs.

### C. Design of Agents

We compared two systems of three agents acting on the MDP described above: a MADDPG system (as described in [18]), and a system using a genetic algorithm to optimize the parameters of the deep neural networks of the agents. The MADDPG agents each had two feed-forward neural networks, one for the actor and one for the critic. The actor networks' inputs were complete observations of the state described above, while their outputs were the actions as described above. The critic networks' inputs were a complete observation of the state plus the actions of *all* agents, while their outputs were the estimated value of the current state. The hidden layers were the same for all networks: two fully connected layers of 64 nodes each, followed by a softmax activation for the actor networks and no activation for the critic networks. We tuned the learning rate to 0.001, discount factor ( $\gamma$ ) to 0.99, and target-network update parameter ( $\tau$ ) to 0.01 for all agents. The training batches were relatively large (256 samples) to mitigate the effects of the observed flat policy gradients. Each training run consisted of 5,000 iterations, each with one data collection episode and three training batches, and the replay buffer was sized to store the transition trajectories for two episodes. The system based on genetic algorithms used identical actor networks to that of the MADDPG system, without the need for critic networks. We optimized the weights of the actor networks with a genetic algorithm, thus eliminating gradient descent. For a tuned population size of 200, we mutated the fittest 10% of each generation using random mutation and a gaussian noise multiplier  $\sigma$  tuned to 0.3 while carrying over the fittest individual unmutated. For both systems, hyperparameter tuning was done through a standard one-at-a-time parameter sweep.

## IV. RESULTS

In Table 1, we show the results of our experiments compared to that of published work on both continuous and discretized action spaces for stock portfolio optimization.

TABLE 1 RESULTS FROM OUR GENETIC ALGORITHM (GA) AND MADDPG SYSTEMS COMPARED TO TYPICAL DDPG AND DQL SYSTEMS.

System	Returns*	Sharpe ratio*	ESG*
Our GA	70.4% $\pm$ 6.8%	3.15 $\pm$ 0.22	26.9 $\pm$ 1.2
Our MADDPG	27.9% $\pm$ 9.5%	1.28 $\pm$ 0.42	29.6 $\pm$ 0.3
DDPG	25% [22]	0.6 [23]	-
DQL	74.9% [25]	3.2 [24]	-

\*Ranges are for 95% confidence intervals.

Our MADDPG returns were in line with the returns reported in the single agent DDPG system in [22], while the

returns of the better performing DQL system in [25] were within the 95% confidence interval of our GA system, making them essentially the same. Further, our GA system outperformed our own MADDPG system in terms of sustainability, with a superior ESG score of 26.9 compared to 29.6. Our GA system also achieved low risk, with a Sharpe ratio of 3.15 which is typically considered “excellent”, while a Sharpe ratio of 1.28 as achieved by the MADDPG system is considered merely “good” [8]. Finally, while [24] reported a similar Sharpe ratio to our system, it was merely a reported metric whereas our system took an active approach to minimizing risk. Our system could thus match [24] in terms of risk *and* [25] in terms of profit, while each of these systems were inferior to ours otherwise. Therefore, though we did not strictly outperform DQL systems in terms of pure financial returns, the fact that we can match their financial returns while offering reduced risk and a sustainable portfolio leads us to claim that our solution is an improvement over the state-of-the-art.

In Fig. 2 we show two typical portfolios held by the two systems during testing. While the GA system quickly achieved a positive portfolio value, the MADDPG system fluctuated around the break-even line for the first half of the episode. Only when the market entered a bullish state after roughly 180 days did we observe a markable increase in the MADDPG portfolio value. This increase was observed much earlier for the GA system – after about 100 days. The fact that the GA system held positions in the GS stock during evaluation, despite this stock having had a mostly downward trend in the training data, suggests that it had learned to interpret market signals as opposed to simply holding the stock that performed best during training. The GA system also responded better to market fluctuations by, for example, taking a position in PG while GS showed bearish signals between ca. 40 and 60 days and choosing to hold cash at times when the MADDPG system did not. The two systems had

clearly learned different strategies, reiterating that for at least one of them the optimum policy remained elusive.

We verified that the substantial difference in performance between the MADDPG and GA systems was due to the nature of the MADDPG system’s policy gradients. In Equation (6) we showed that the objective function of a MADDPG system is expressed in terms of the parameters of the actor ( $\mu$ ) and critic ( $Q$ ) networks. Fig. 3 illustrates the steepest negative gradients of each of the actor and critic networks during training and gives an indication of how well an optimizer may perform at gradient descent; if all gradients are flat – or close to zero – then the optimizer has no indication of how to adjust the weights. From this figure, we observed that while the three *critics* appeared to have had sufficient gradients to perform gradient descent, the *actors* all experienced flat gradients. This suggests that the critics were able to learn the values of states, but the agents were not able to effectively use these values to find optimum policies. This might be due to the critics having had a more holistic view of the state action space, as intended by the authors of MADDPG [18]. We therefore conclude that the optimum policy had remained elusive to the MADDPG system, as substantiated by the higher returns achieved using the GA system.

Finally, in Fig. 4 and Fig. 5, we show the actions taken by the *individual agents* of the two systems. The agents of the GA system clearly took more distinct roles, with the risk-averse agent frequently voting to hold cash and generally avoiding the most volatile of the three stocks (GS), which the profit agent mostly favored. Interestingly, this clear separation of responsibility was not evident in the behavior of the MADDPG agents which acted more haphazardly. In future work, we intend to more closely inspect the behaviors of the agents and we aim to characterize them and extract explanations for their actions.

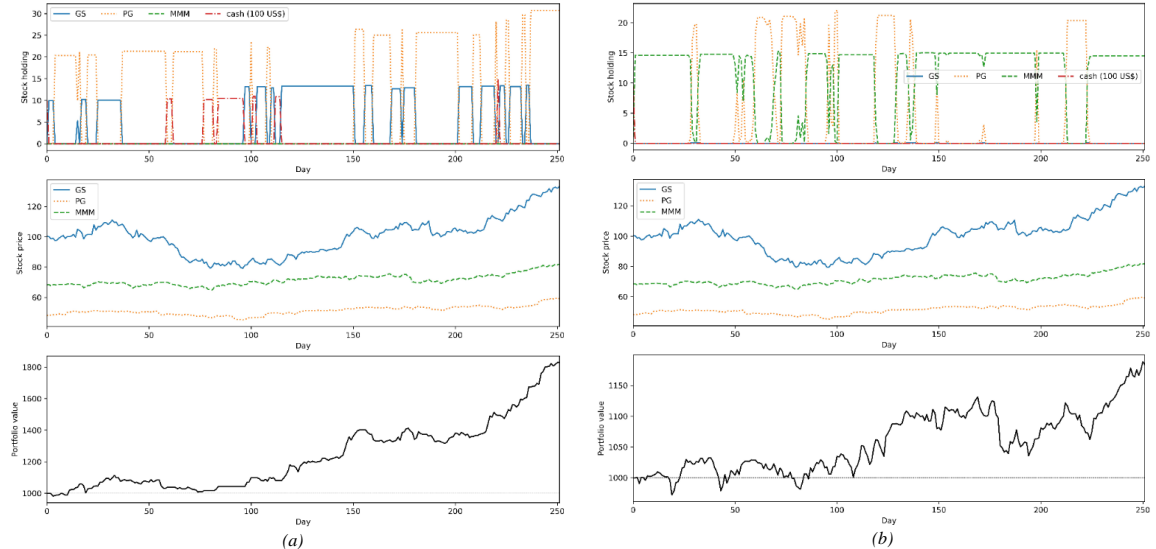


Fig. 2 Two typical portfolios when trading stocks on test data using (a) the genetic algorithm system and (b) the MADDPG system of traders. The upper row shows the daily positions of assets – GS, PG, MMM, and cash – for each system, the middle row shows the daily close prices, and the bottom row shows the total portfolio values of each system. While the system in (a) hardly has negative portfolio values and finally achieves a return of 83% (Sharpe ratio = 3.4 and ESG = 26.0), the system in (b) frequently has negative portfolio values in early stages and finally yields a return of 19% (Sharpe ratio = 1.2 and ESG = 27.8.)

## V. CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

In this study, we defined the problem of stock portfolio optimization in terms of reinforcement learning and designed a multi-agent system with a continuous action space. From published works we showed that, for stock portfolio optimization, DQL has typically been outperforming policy gradient methods despite being limited to discretized actions. We then showed that, for our problem, this was due to flat policy gradients inhibiting gradient descent from finding the optimum policy. Since continuous action spaces are nevertheless considered preferable to discretization in stock portfolio optimization, we overcame the policy gradient problem by replacing gradient descent with a genetic algorithm for parameter optimization. We showed that this method outperformed MADDPG in a three-agent system trading a selection of three stocks from the DOW30 index. Furthermore, our agents were rewarded not only for financial returns, but also for risk (via the Sharpe ratio), and sustainability (via ESG). While risk is key in modern portfolio theory, sustainability is increasingly becoming requisite in modern trading strategies. It is therefore pivotal that state-of-the-art solutions not only support reporting of such metrics, but that they actively optimize portfolios to improve on them. Our main contribution is therefore the inclusion of the Sharpe ratio and ESG in the utility function for portfolio optimization, while matching state-of-the-art solutions in terms of financial returns. We claim that is not necessary to outperform current solutions in terms of financial returns since our solution offers the same returns with reduced risk in a sustainable portfolio, making it an improvement on the state-of-the-art.

Our ultimate objective is the development of personalized digital financial advisors using customer micro-segmentation [28]. These financial advisors will recommend, in an explainable way, an optimum allocation of funds given a personal budget and a portfolio of financial products and services. We therefore intend to address the explainability of our system in future work by characterizing, explaining, and predicting an agent's policy based on the history of past trades.

## ACKNOWLEDGMENT

We acknowledge Phillip Tabor for his implementation of the MADDPG algorithm. Some of the code for this paper was adapted from his library.

## REFERENCES

- [1] Y. Liu, Q. Liu, H. Zhao, Z. Pan and C. Liu, "Adaptive quantitative trading: An imitative deep reinforcement learning approach," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 2, pp. 2128-2135, 2020.
- [2] T. G. Fischer, "Reinforcement learning in financial markets - a survey," *FAU Discussion Papers in Economics*, vol. 12, no. 1, pp. 1-46, 2018.
- [3] G. Huang, X. Zhou and Q. Song, "Deep reinforcement learning for portfolio management based on the empirical study of chinese stock market," *arXiv*, vol. 2012.13773, pp. 1-37, 2020.
- [4] X. Y. Liu, H. Yang, Q. Chen, R. Zhang, L. Yang, B. Xiao and C. D. Wang, "FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance," *arXiv*, vol. 2011.09607v1, pp. 1-11, 2020.
- [5] Z. Jiang, D. Xu and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *arXiv*, vol. 1706.10059, pp. 1-31, 2017.
- [6] A. Mosavi, Y. Faghan, P. Ghamisi, P. Duan, F. S. Ardabili, E. Salwana and S. S. Band, "Comprehensive review of deep reinforcement learning methods and applications in economics," *Mathematics*, vol. 8, no. 10, pp. 1640-1682, 2020.
- [7] F. S. Willaim, "The Sharpe ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49-58, 1994.
- [8] J. B. Maverick, "Investopedia," 30 04 2021. [Online]. Available: <https://www.investopedia.com/ask/answers/010815/what-good-sharpe-ratio.asp>. [Accessed 30 06 2021].
- [9] M.-E. Wu, J.-H. Syu, J. C.-W. Lin and J.-M. Ho, "Portfolio management system in equity market neutral using reinforcement learning," *Applied Intelligence*, vol. 51, no. 9, pp. 1-13, 2021.
- [10] JP Morgan Chase & Co., "Environmental social & governance report," JP Morgan Chase & Co., 2020.
- [11] Wells Fargo & Co., "Environmental, social, and governance (ESG) report," Wells Fargo & Co., 2020.
- [12] Goldman Sachs, "The future now: Integrating sustainability with purpose across our business," Goldman Sachs, 2020.
- [13] T. T.-L. Chong and W.-K. Ng, "Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30," *Applied Economics Letters*, vol. 15, no. 14, pp. 1111-1114, 2008.
- [14] T. T.-L. Chong, W.-K. Ng and V. K.-S. Liew, "Revisiting the performance of MACD and RSI oscillators," *Journal of Risk and Financial Management*, vol. 7, no. 1, pp. 1-12, 2014.
- [15] L. Goldberg and O. Mahmoud, "Drawdown: From practice to theory and back again," *Mathematics and Financial Economics*, vol. 11, no. 3, pp. 275-297, 2017.
- [16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press, 2018.
- [17] R. E. Bellman, *Dynamic Programming*, Princeton: Princeton University Press, 1957.
- [18] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in Neural Information Processing Systems*, vol. 30, no. 1, pp. 1-12, 2017.
- [19] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," *arXiv*, vol. 1712.06567, pp. 1-16, 2017.
- [20] A. Sehgal, H. La, S. Louis and H. Nguyen, "Deep reinforcement learning using genetic algorithm for parameter optimization," in *Third IEEE International Conference on Robotic Computing (IRC)*, Naples, Italy, 2019.
- [21] H. Markovitz, "Portfolio Selection," *Journal of Finance*, vol. 7, no. 1, pp. 77-91, 1952.
- [22] L. T. Hieu, "Deep reinforcement learning for stock portfolio optimization," *International Journal of Modeling and Optimization*, vol. 10, no. 5, pp. 139-144, 2020.
- [23] H. Zhang, Z. Jiang and J. Su, "A deep deterministic policy gradient-based strategy for stocks portfolio management," *arXiv*, vol. 2103.11455v1, pp. 1-8, 2021.
- [24] G. Lucarelli and M. Borrotti, "A deep Q-learning portfolio management framework for the cryptocurrency market," *Neural Computing and Applications*, vol. 32, no. 1, pp. 17229-17244, 2020.
- [25] J. W. Lee, J. Park, J. O. J. Lee and E. Hong, "A multiagent approach to Q-Learning for daily stock trading," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 37, no. 6, pp. 864-877, 2007.
- [26] K. Cobbe, O. Klimov, C. Hesse, T. Kim and J. Schulman, "Quantifying generalization in reinforcement learning," *arXiv*, vol. 1812.02341v3, no. 1, pp. 1-14, 2019.
- [27] C. Zhang, O. Vinyals, R. Munos and S. Bengio, "A study on overfitting in deep reinforcement learning," *arXiv*, vol. 1804.06893v1, no. 1, pp. 1-19, 2018.
- [28] C. Maree and C. W. Omlin, "Clustering in recurrent neural networks for micro-segmentation using spending personality," *IEEE Symposium Series on Computational Intelligence*, pp. 1-5, 2021.
- [29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, "Playing Atari with deep reinforcement learning," *arXiv*, vol. 1312.5602, pp. 1-9, 2013.

## VI. APPENDIX

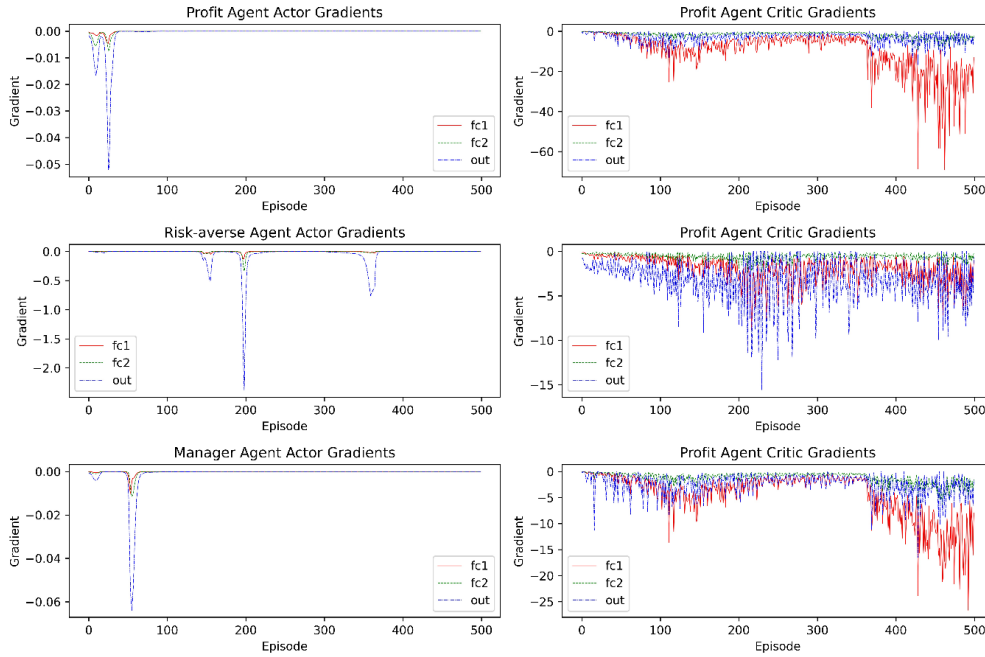


Fig. 3 The steepest policy gradients of the three MADDPG agents' actor and critic networks for the first 500 training episodes. Each datapoint shows the largest negative component of the gradients of the weights for each of the fully connected (fc1, fc2) and output layers (out). While the critic networks have workable gradients, the gradients for the actor networks are mostly flat throughout training.

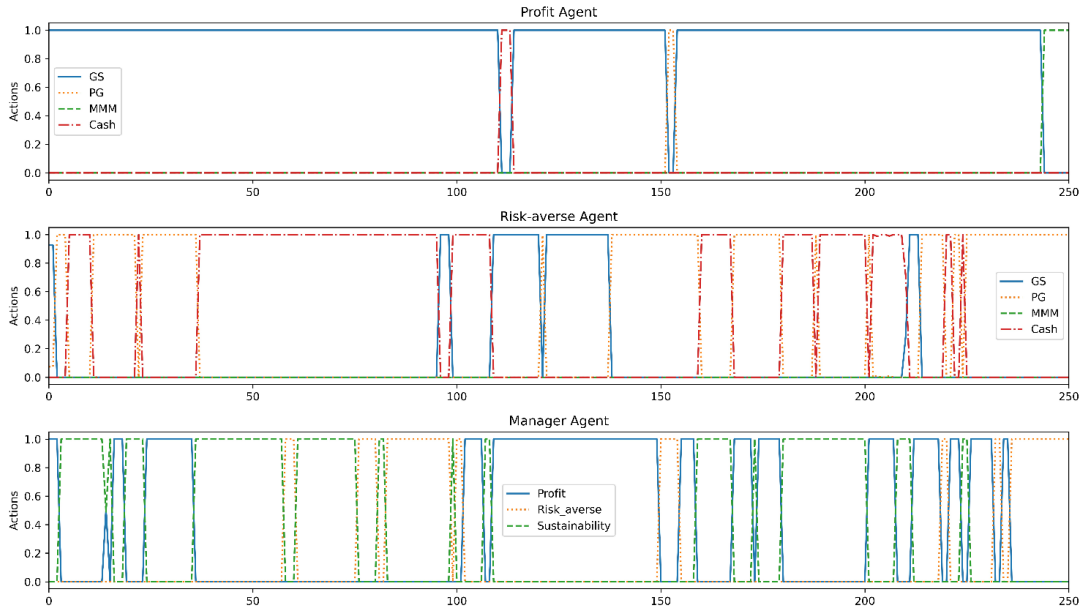


Fig. 4 The agents' actions in a typical GA system during testing. The first plot shows the profit agent frequently voting to hold GS, while the second plot shows the risk-averse agent more most frequently voting to holding cash. The manager agent's role was to choose the weighting between the other agents' votes; the last plot shows it frequently varying between all three objectives: profit, risk, and sustainability.

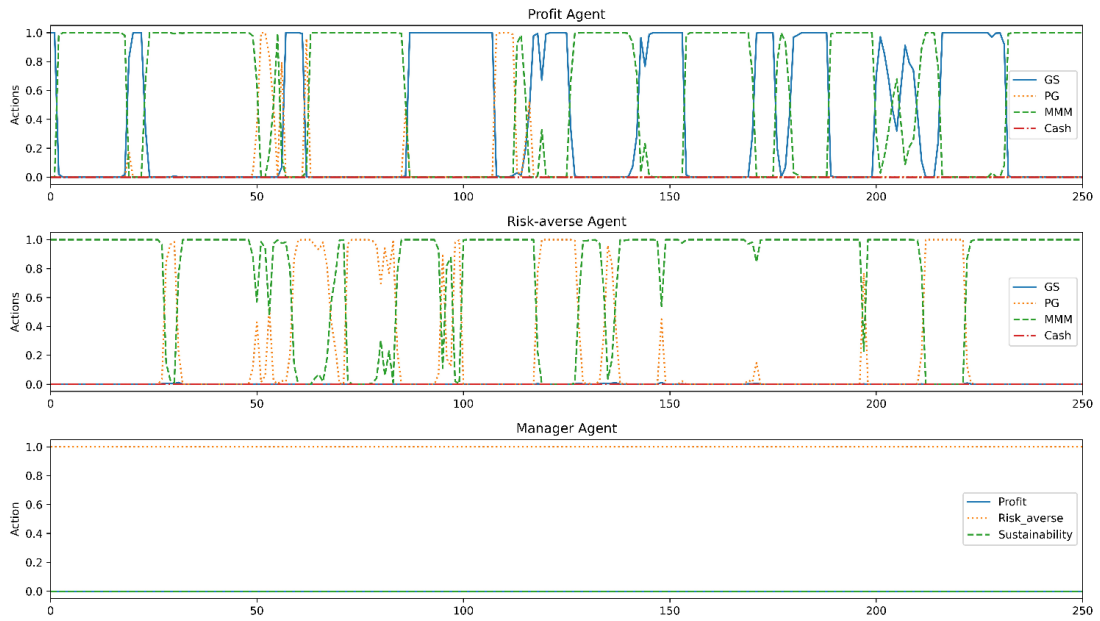


Fig. 5 The agents' actions in a typical MADDPG system during testing. The first plot shows the profit agent's vote varying mostly between GS and MMM, while the second plot shows the risk-averse agent voting mostly for MMM, interestingly without ever holding cash. The manager agent's role was to choose the weighting between the other agents' votes; the last plot shows that it always chose to accept the vote of the risk-averse agent.



## Appendix E

# Reinforcement Learning Your Way: Agent Characterization through Policy Regularization

This paper has been published as:

C. Maree and C. W. Omlin, “Reinforcement Learning Your Way: Agent Characterization through Policy Regularization”, *AI*, **2022**, 3(2), pp. 250–259, doi: 10.3390/ai3020015.

Copyright © Creative Commons Attribution (CC BY)



Article

# Reinforcement Learning Your Way: Agent Characterization through Policy Regularization

Charl Maree <sup>1,2,\*</sup>  and Christian Omlin <sup>2,†</sup>

<sup>1</sup> Chief Technology Office, Sparebank 1 SR-Bank, 4007 Stavanger, Norway

<sup>2</sup> Center for AI Research, University of Agder, 4879 Grimstad, Norway; christian.omlin@uia.no

\* Correspondence: charl.maree@uia.no

† Current address: Jon Lilletunsvet 9, 4879 Grimstad, Norway.

**Abstract:** The increased complexity of state-of-the-art reinforcement learning (RL) algorithms has resulted in an opacity that inhibits explainability and understanding. This has led to the development of several post hoc explainability methods that aim to extract information from learned policies, thus aiding explainability. These methods rely on empirical observations of the policy, and thus aim to generalize a characterization of agents' behaviour. In this study, we have instead developed a method to imbue agents' policies with a characteristic behaviour through regularization of their objective functions. Our method guides the agents' behaviour during learning, which results in an intrinsic characterization; it connects the learning process with model explanation. We provide a formal argument and empirical evidence for the viability of our method. In future work, we intend to employ it to develop agents that optimize individual financial customers' investment portfolios based on their spending personalities.

**Keywords:** explainable AI; multi-agent systems; deterministic policy gradients



Citation: Maree, C.; Omlin, C.

Reinforcement Learning Your Way: Agent Characterization through Policy Regularization. *AI* **2022**, *3*, 250–259. <https://doi.org/10.3390/ai3020015>

Academic Editor: Gianni D'Angelo

Received: 20 January 2022

Accepted: 22 March 2022

Published: 24 March 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recent advances in reinforcement learning (RL) have increased complexity which, especially for deep RL, has brought forth challenges related to explainability [1]. The opacity of state-of-the-art RL algorithms has led to model developers having a limited understanding of their agents' policies and no influence over learned strategies [2]. While concerns surrounding explainability have been noted for AI in general, it is only more recently that attempts have been made to explain RL systems [1,3–5]. These attempts have resulted in a wide suite of methods that typically rely on post hoc analysis of learned policies, which give only observational assurances of agents' behaviour. However, it is pivotal that future development of RL methods focus on more fundamental approaches towards inherently explainable RL [1]. We therefore propose an *intrinsic* method of guiding an agent's learning by controlling the objective function; there are two ways of manipulating the learning objective: modifying the reward function and regularizing the actions taken during learning. Whereas the reward function is specific to the particular problem, our ambition is to establish a generic method. We therefore propose a method which regularizes the objective function by minimizing the difference between the observed action distribution and a desired prior action distribution; we thus bias the actions that agents learn. While current methods for RL regularization aim to improve *training performance*—e.g., by maximizing the entropy of the action distribution [6], or by minimising the distance to a prior sub-optimal state-action distribution [7]—our aim is the *characterization* of our agents' behaviours. We extend single-agent regularization to accommodate multi-agent systems, which allows intrinsic characterization of individual agents. We provide a formal argument for the rationale of our method and demonstrate its efficacy in a toy problem where agents learn to navigate to a destination on a grid by performing, e.g., only right turns (under the premise that right turns are considered safer than left turns [8]). There are

several useful applications beyond this toy problem, such as asset management based on personal goals, intelligent agents with intrinsic virtues, and niche recommender systems based on customer preferences.

## 2. Background and Related Work

### 2.1. Agent Characterization

There have been several approaches to characterizing RL agents, with most—if not all—employing some form of post hoc evaluation technique. Some notable examples are:

*Probabilistic argumentation* [9] in which a human expert creates an ‘argumentation graph’ with a set of arguments and sub-arguments; sub-arguments attack or support main arguments which attack or support discrete actions. Sub-arguments are labelled as ‘ON’ or ‘OFF’ depending on the state observation for each time-step. Main arguments are labelled as ‘IN’, ‘OUT’, or ‘UNDECIDED’ in the following RL setting: *states* are the union of the argumentation graph and the learned policy, *actions* are the probabilistic ‘attitudes’ towards given arguments, and *rewards* are based on whether an argument attacks or supports an action. The learned ‘attitudes’ towards certain arguments are used to characterize agents’ behaviour.

*Structural causal modelling (SCM)* [10] learns causal relationships between states and actions through ‘action influence graphs’ that trace all possible paths from a given initial state to a set of terminal states, via all possible actions in each intermediate state. The learned policy then identifies a causal chain as the single path in the action influence graph that connects the initial state to the relevant terminal state. The explanation is the vector of rewards along the causal chain. Counter-explanations are a set of comparative reward vectors along chains originating from counter-actions in the initial state. Characterizations are made based on causal and counterfactual reasons for agents’ choice of action.

*Reward decomposition* [11,12] decomposes the reward into a vector of intelligible reward classes using expert knowledge. Agent characterization is done by evaluation of the reward vector for each action post training.

*Hierarchical reinforcement learning (HRL)* [13,14] divides agents’ tasks into sub-tasks to be learned by different agents. This simplifies the problem to be solved by each agent, making their behaviour easier to interpret, and thereby making them easier to characterize.

*Introspection (interesting elements)* [15] is a statistical post hoc analysis of the policy. It considers elements such as the frequency of visits to states, the estimated values of states and state-action pairs, state-transition probabilities, how much of the state space is visited, etc. Interesting statistical properties from this analysis are used to characterize the policy.

### 2.2. Multi-Agent Reinforcement Learning and Policy Regularization

We consider the multi-agent setting of partially observable Markov decision processes (POMDPs) [16]: for  $N$  agents, let  $\mathcal{S}$  be a set of states,  $\mathcal{A}_i$  a set of actions, and  $\mathcal{O}_i$  a set of incomplete state observations where  $i \in [1, \dots, N]$  and  $\mathcal{S} \mapsto \mathcal{O}_i$ . Agents select actions according to individual policies  $\pi_{\theta_i}(\mathcal{O}_i) \mapsto \mathcal{A}_i$  and receive rewards according to individual reward functions  $r_i(\mathcal{S}, \mathcal{A}_i) \mapsto \mathbb{R}$ , where  $\theta_i$  is the set of parameters governing agent  $i$ ’s policy. Finally, agents aim to maximize their total discounted rewards:

$$R_i(o, a) = \sum_{t=0}^T \gamma r_i(o_t, a_t)$$

where  $T$  is the time horizon and  $\gamma \in [0, 1]$  is a discount factor. For single-agent systems, the deep deterministic policy gradient algorithm (DDPG) defines the gradient of the objective  $J(\theta) = \mathbb{E}_{s \sim p^\mu} [R(s, a)]$  as [17]:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim \mathcal{D}} \left[ \nabla_{\theta} \mu_{\theta}(a|s) \nabla_a Q^{\mu_{\theta}}(s, a) \Big|_{a=\mu_{\theta}(s)} \right] \quad (1)$$

where  $p^{\mu}$  is the state distribution,  $\mathcal{D}$  is an experience replay buffer storing observed state transition tuples  $(s, a, r, s')$ , and  $Q^{\mu_{\theta}}(s, a)$  is a state-action value function where actions are selected according to a policy  $\mu_{\theta}(\mathcal{S}) \mapsto \mathcal{A}$ . In DDPG, the policy  $\mu$ —also called the *actor*—and the value function  $Q$ —also called the *critic*—are modelled by deep neural networks. Equation (1) is extended to a multi-agent setting; the multi-agent deep deterministic policy gradient algorithm (MADDPG) learns individual sets of parameters for each agent  $\theta_i$  [18]:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{o_i, a \sim \mathcal{D}} \left[ \nabla_{\theta_i} \mu_{\theta_i}(a_i | o_i) \nabla_{a_i} Q^{\mu_{\theta_i}}(o_i, a_1, \dots, a_N) \Big|_{a_i = \mu_{\theta_i}(o_i)} \right] \quad (2)$$

where  $o_i \in \mathcal{O}_i$  and the experience replay buffer  $\mathcal{D}$  contains tuples  $(o_i, a_i, r_i, o'_i)$ ,  $i \in [1, \dots, N]$ .

In this work, we further extend MADDPG by adding a regularization term to the actors' objective functions, thus encouraging them to mimic the behaviours specified by simple predefined prior policies. There have been several approaches to regularizing RL algorithms, mostly for the purpose of improved generalization or training performance. In [7], the authors defined an objective function with a regularization term related to the statistical difference between the current policy and a predefined prior:

$$J(\theta) = \mathbb{E}_{s, a \sim \mathcal{D}} [R(s, a) - \alpha D_{KL}(\pi_{\theta}(s, a) \| \pi_0(s, a))] \quad (3)$$

where  $\alpha$  is a hyperparameter scaling the relative contribution of the regularization term—the Kullback–Leibler (KL) divergence ( $D_{KL}$ )—and  $\pi_0$  is the prior policy which the agent attempts to mimic while maximising the reward. The KL divergence is a statistical measure of the difference between two probability distributions, formally:

$$D_{KL}(P \| Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

where  $P$  and  $Q$  are discrete probability distributions on the same probability space  $X$ . The stated objective of KL regularization is increased learning performance by penalising policies that stray too far from the prior. The KL divergence is often also called the *relative entropy*, with KL-regularized RL being the generalization of entropy-regularized RL ([19]); specifically if  $\pi_0$  is the uniform distribution, Equation 3 reduces to, up to a constant, the objective function for entropy-regulated RL as described in [6]:

$$J(\theta) = \mathbb{E}_{s, a \sim \mathcal{D}} [R(s, a) + \alpha H[\pi_{\theta}(s, a)]] \quad (4)$$

where  $H(\pi) = P(\pi) \log(P(\pi))$  is the statistical entropy of the policy. The goal of entropy-regularized RL is to encourage exploration by maximising the policy's entropy and is used as standard in certain state-of-the-art RL algorithms, such as soft actor-critic (SAC) [6]. Other notable regularization methods include *control regularization* where, during learning, the action of the actor is weighted with an action from a sub-optimal prior:  $\mu_k = \frac{1}{1+\lambda} \mu_{\theta} + \frac{\lambda}{1+\lambda} \mu_{\text{prior}}$  and *temporal difference regularization*, which adds a penalty for large differences in the Q-values of successive states:  $J(\theta, \eta) = \mathbb{E}_{s, a, s' \sim \mathcal{D}} [R(s, a) - \eta \delta_Q(s, a, s')]$ , where  $\delta_Q(s, a, s') = [R(s, a) + \gamma Q(s', a) - Q(s, a)]^2$  [20,21].

While our algorithm is based on regularization of the objective function, it could be argued that it shares similar goals as those of algorithms based on *constrained RL*, namely the intrinsic manipulation of agents' policies towards given objectives. One example of constrained RL is [22], which finds a policy whose long-term measurements lie within a set of constraints by penalising the reward function with the Euclidean distance between the state and a given set of restrictions, e.g., an agent's location relative to obstacles on a map. Another example is [23], which penalises the value function with the accumulated cost of a series of actions, thus avoiding certain state-action situations. However, where constrained RL attempts to *avoid* certain conditions—usually through a penalty based on expert knowledge of the state—regularized RL aims to *promote* desired behaviours, such as choosing default actions during training or maximizing exploration by maximising

action entropy. The advantage of our system is that it does not require expert knowledge of the state-action space to construct constraints; our regularization term is independent of the state, which allows agents to learn simple behavioural patterns, thus improving the interpretability of their characterization.

### 3. Methodology

We regulate our agents based on a state-independent prior to maximize rewards while adhering to simple, predefined rules. In a toy problem, we demonstrate that agents learn to find a destination on a map by taking only right turns. Intuitively, we supply the probability distribution of three actions—left, straight, and right—as a regularization term in the objective function, meaning the agents aim to mimic this given probability distribution while maximizing rewards. Such an agent can thus be characterized as an agent that prefers, e.g., right turns over left turns. As opposed to post hoc characterization, ours is an intrinsic method that inserts a desirable characteristic into an agent’s behaviour *during* learning.

#### Action Regularization

We modify the objective function in Equation (4) and replace the regularization term  $H[\pi_\theta(s, a)] = P(\pi) \log(P(\pi))$  with the mean squared error of the expected action and a specified prior  $\pi_0$ :

$$J(\theta) = \mathbb{E}_{s, a \sim \mathcal{D}}[R(s, a)] - \lambda L \quad (5)$$

$$L = \frac{1}{M} \sum_{j=0}^M \left[ \mathbb{E}_{a \sim \pi_\theta} [a_j] - (a_j | \pi_0(a)) \right]^2 \quad (6)$$

where  $\lambda$  is a hyperparameter that scales the relative contribution of the regularization term  $L$ ,  $a_j$  is the  $j^{\text{th}}$  action in a vector of  $M$  actions,  $\pi_\theta$  is the current policy, and  $\pi_0$  is the specified prior distribution of actions, which the agent aims to mimic while maximising the reward. Note that  $\pi_0(a)$  is *independent* of the state and  $(a_j | \pi_0(a))$  is therefore constant across all observations and time-steps. This is an important distinction from previous work, and results in a prior that is simpler to construct and a characterization that can be interpreted by non-experts; by removing the reference to the state space, we reduce the interpretation to the action space only, i.e., in this example the agent either proceeds straight, turns left, or turns right, independent of the locations of the agent and destination. Since this is a special case of Equation (4), it follows from the derivation given in [6].

We continue by extending our objective function to support a multi-agent setting. From Equation (5) and following the derivation in [18], we derive a multi-agent objective function with  $i \in [1, N]$  where  $N$  is the number of agents:

$$J(\theta_i) = \mathbb{E}_{o_i, a_i \sim \mathcal{D}}[R_i(o_i, a_i)] - \lambda \frac{1}{M_i} \sum_{j=0}^{M_i} \left[ \mathbb{E}_{a \sim \pi_{\theta_i}(o_i, a)} (a_j) - (a_j | \pi_{0_i}(a)) \right]^2 \quad (7)$$

Further, in accordance with the MADDPG algorithm, we model actions and rewards with actors and critics, respectively [18]:

$$\mathbb{E}_{a_i \sim \pi_{\theta_i}(o_i, a_i)} [a_i] = \mu_{\theta_i}(o_i) \quad (8)$$

$$\mathbb{E}_{o_i, a_i \sim \mathcal{D}} [R_i(o_i, a_i)] = Q_{\theta_i}(o_i, \mu_{\theta_1}(o_1), \dots, \mu_{\theta_N}(o_N)) \quad (9)$$

Through simple substitution of Equations (8) and (9) into Equation (7), we formulate our multi-agent regularized objective function:

$$J(\theta_i) = Q_{\theta_i}(o_i, \mu_{\theta_1}(o_1), \dots, \mu_{\theta_N}(o_N)) - \lambda L_i \quad (10)$$

$$L_i = \frac{1}{M_i} \sum_{j=0}^{M_i} \left[ \mu_{\theta_i}(o_i)_j - (a_j | \pi_{0_i}(a)) \right]^2 \quad (11)$$

Algorithm 1 optimizes the policies of multiple agents given individual regularization constraints  $\pi_{0_i}$ .

---

**Algorithm 1** Action-regularized MADDPG algorithm.
 

---

```

Set the number of agents  $N \in \mathbb{N}$ 
for  $i$  in  $1, N$  do ▷ For each agent
  Initialize actor network  $\mu_{\theta_{\mu,i}}$  with random parameters  $\theta_{\mu,i}$ 
  Initialize critic network  $Q_{\theta_{Q,i}}$  with random parameters  $\theta_{Q,i}$ 
  Initialize target actor network  $\mu_{\theta'_{\mu,i}}$  with parameters  $\theta'_{\mu,i} \leftarrow \theta_{\mu,i}$ 
  Initialize target critic network  $Q_{\theta'_{Q,i}}$  with parameters  $\theta'_{Q,i} \leftarrow \theta_{Q,i}$ 
  Set the desired prior action distribution  $\pi_{0_i}$ 
  Set the number of actions  $M_i \leftarrow |\pi_{0_i}|$ 
end for
Initialize replay buffer  $\mathcal{D}$ 
Set regularization weight  $\lambda$ 
for  $e = 1, \text{Episodes}$  do
  Initialise random function  $F(e) \sim \mathcal{N}(0, \sigma_e)$  for exploration
  Reset environment and get the state observation  $s_1 \mapsto o_{[1..N]}$ 
   $t \leftarrow 1, \text{Done} \leftarrow \text{False}$ 
  while not Done do
    for  $i$  in  $1, N$  do ▷ For each agent
      Select action with exploration  $a_{i,t} \leftarrow \mu_{\theta_{\mu,i}}(o_i) + F(e)$ 
    end for
    Apply compounded action  $a_t$ 
    Retrieve rewards  $r_{[1..N],t}$  and observations  $s'_t \mapsto o'_{[1..N],t}$ 
    Store transition tuple  $\mathcal{T} = (o_t, a_t, r_t, o'_t)$  to replay buffer:  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{T}$ 
     $t \leftarrow t + 1$ 
    if (end of episode) then
       $\text{Done} \leftarrow \text{True}$ 
    end if
  end while
  Sample a random batch from the replay buffer  $\mathcal{B} \subset \mathcal{D}$ 
  for  $i$  in  $1, N$  do ▷ For each agent
     $\widehat{Q}_i \leftarrow r_{\mathcal{B},i} + \gamma Q'_i(o'_{\mathcal{B},i}, \mu'_i(o'_{\mathcal{B},1}), \dots, \mu'_N(o'_{\mathcal{B},N}))$ 
    Update critic parameters  $\theta_{Q,i}$  by minimising the loss:
    
$$\mathcal{L}(\theta_{Q,i}) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \left( Q_{\theta_{Q,i}}(o_{\mathcal{B}}, a_{\mathcal{B},1}, \dots, a_{\mathcal{B},N}) - \widehat{Q}_i \right)^2$$

    Update the actor parameters  $\theta_{\mu,i}$  by minimising the loss: ▷ From Equation (10)
    
$$\widehat{R}_i = \overline{Q_i(o_{\mathcal{B},i}, \mu_1(o_{\mathcal{B},1}), \dots, \mu_N(o_{\mathcal{B},N}))}$$

    
$$\mathcal{L}(\theta_{\mu,i}) = -\widehat{R}_i + \lambda \frac{1}{M_i} \sum_{j=1}^{M_i} \left[ \mu_i(o_{\mathcal{B},i})_j - (a_j | \pi_{0_i}) \right]^2$$

    Update target network parameters:
    
$$\theta'_{\mu,i} \leftarrow \tau \theta_{\mu,i} + (1 - \tau) \theta'_{\mu,i}$$

    
$$\theta'_{Q,i} \leftarrow \tau \theta_{Q,i} + (1 - \tau) \theta'_{Q,i}$$

  end for
end for

```

---

## 4. Experiments

### 4.1. Empirical Setup

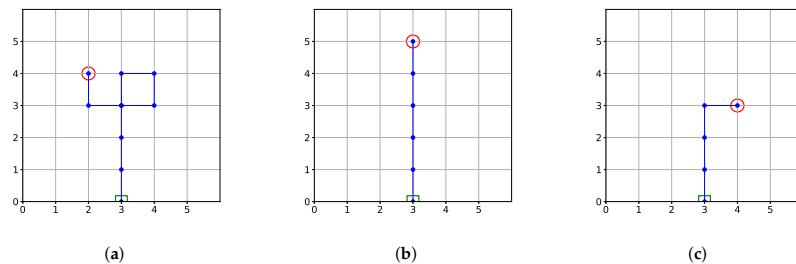
We created a toy problem in which one or more agents navigate a  $6 \times 6$  grid through a set of three actions:  $\mathcal{A}_1 = \text{turn left}$ ,  $\mathcal{A}_2 = \text{go straight}$ , and  $\mathcal{A}_3 = \text{turn right}$ . Every new episode randomly placed a set of destinations in the grid  $D_i$ ,  $i \in [1, N]$ —one for each of  $N \geq 1$  agents—with initial agent locations  $L_{i,0} = (3, 0)$ . Rewards were the agents' Euclidean

distances from their destinations  $R_{i,t} = \|D_i - L_{i,t}\|_2$  where  $L_{i,t}$  is the location of agent  $i$  at time-step  $t$ . Finally, agents' observations were the two-dimensional distances to their destinations:  $\mathcal{O}_{i,t} = D_i - L_{i,t}$ . An episode was completed when either both agents had reached their destinations or a maximum of 50 time steps had passed.

We ran two sets of experiments, one for a single-agent setting and one for a dual-agent setting. We sized all networks in these two settings with two fully connected feed-forward layers; the single agent networks had 200 nodes in each layer, while the dual-agent networks had 700 nodes in each layer. Actor networks had a softmax activation layer, while the critic networks remained unactivated. Our training runs consisted of 3000 iterations and we tuned the hyperparameters using a simple one-at-a-time parameter sweep. We used training batches of 256 time-steps and sized the reply buffers to hold 2048 time-steps. In each iteration, we collected 256 time steps and ran two training epochs. We tuned the learning rates to 0.04 for the actors and 0.06 for the critics, the target network update parameters  $\tau$  to 0.06, and the discount factors  $\gamma$  to 0.95. We specified the regularization coefficient  $\lambda = 2$ , the regularization prior for the single-agent setting as  $\pi_0 = [P(\mathcal{A}_1), P(\mathcal{A}_2), P(\mathcal{A}_3)] = [0.0, 0.6, 0.4]$ , and the regularization priors for the dual-agent setting as  $\pi_{0,1} = [0.0, 0.6, 0.4]$  and  $\pi_{0,2} = [0.4, 0.6, 0.0]$ . This meant that the single agent was regularized to not take any left turns, while slightly favouring going straight above turning right. For the dual agents, agent 1 was to avoid left turns while agent 2 was to instead avoid right turns; we did this to demonstrate the characterization of the agents as preferring either left or right turns while navigating to their destinations. We conducted three experiments to explore the effects of the regularization prior, using the single-agent system for brevity, with the regularization priors  $\pi_0 \in \{[0.4, 0.2, 0.4], [0.33, 0.33, 0.33], [0.1, 0.5, 0.4]\}$ .

#### 4.2. Results

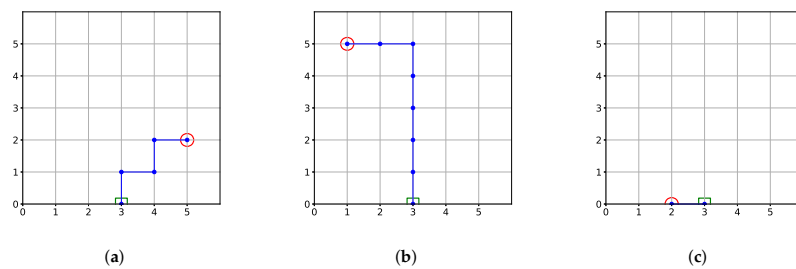
In the single agent setting of our toy problem, we used our algorithm to encourage an agent to prefer right turns over left turns; we used a regularization prior  $\pi_0 = [0.0, 0.6, 0.4]$  to regulate the probability of *left actions* to 0.0, *straight actions* to 0.6, and *right actions* to 0.4. Figure 1 shows three different trajectories that demonstrate such an agent's behaviour for destinations which lie either to the left, straight ahead, or to the right of the agent's starting location. As expected, the agent never turned left and always took the shortest route to its destination given its constraints.



**Figure 1.** Three trajectories of a single agent in the navigation problem. The starting locations are consistently (3,0), and the destinations are marked by red circles. During learning, the agent received a regularization prior  $\pi_0 = [0.0, 0.6, 0.4]$ , where the values in  $\pi_0$  correspond to the probabilities of the actions turn left, go straight, and turn right, respectively. While the agent in (a) makes a series of right turns to reach its destination on the left, the agent in (b) needs not turn, and the agent in (c) follows the shortest path involving a single right turn.

Figure 2 shows three additional experiments which illustrate an agent's behaviour given different regularization priors. In Figure 2a, we used the prior  $\pi_0 = [0.4, 0.2, 0.4]$ , which biased the agent towards taking turns rather than going straight. This agent consistently followed a zig-zag approach to the target, using the same number of steps compared

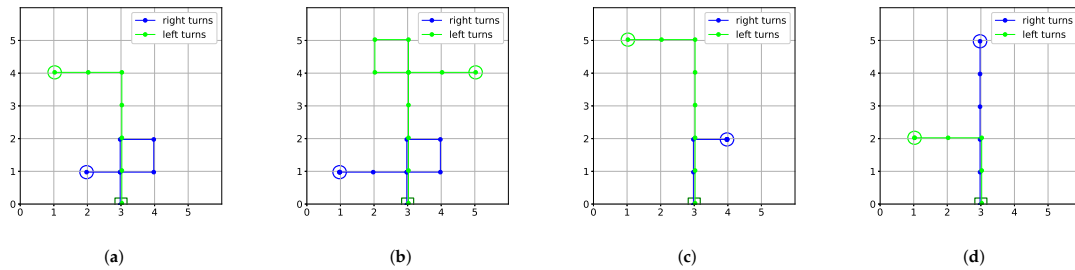
to a direct path with a single turn. This is an interesting observation, as an unregulated agent would typically take a direct path, as shown in Figure 2b. This agent was regulated with a uniform prior  $\pi_0 = [0.33, 0.33, 0.33]$ , which resulted in a similar strategy as that of an unregulated agent, but with the added benefit of increased exploration as discussed in [6]—entropy regularization uses the uniform distribution for  $\pi_0$ . In Figure 2c, we used the prior  $\pi_0 = [0.1, 0.5, 0.4]$  which assigns a low probability for taking left turns. In this experiment we specifically chose a destination to the immediate left of the starting location; other destinations allowed the agent to take the preferred right turns, whereas an immediate left turn to this shown destination proves that the agent does take this action in special cases. These behaviours are also observed in the multi-agent setting which, for brevity, we do not show here.



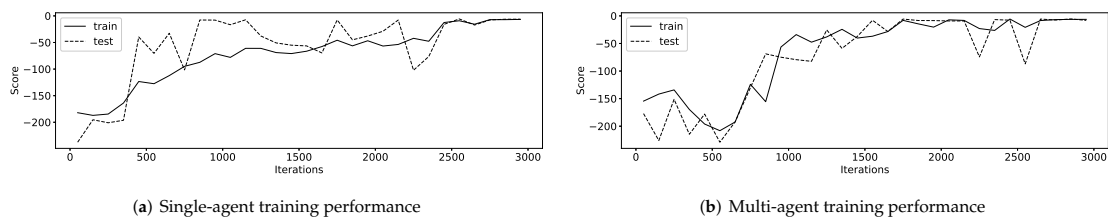
**Figure 2.** Three trajectories of agents trained with various regularization priors. In (a), the agent is biased towards taking turns and follows a zig-zag trajectory towards the destination ( $\pi_0 = [0.4, 0.2, 0.4]$ ). In (b), the agent's regularization prior is uniform—which equally favours all actions—and it follows the shortest path to the destination ( $\pi_0 = [0.33, 0.33, 0.33]$ ). In (c), the agent is allowed to take left turns with a low probability ( $\pi_0 = [0.1, 0.5, 0.4]$ ); we specifically chose the shown destination to encourage the agent to make a left turn.

Figure 3 shows the same grid navigation problem, but this time for a multi-agent setting. Here, we used two agents with different regularization terms to constrain their actions to (1) right turns only and (2) left turns only; the first agent's regularization prior  $\pi_{0,1} = [0.0, 0.6, 0.4]$  specified a probability of 0.0 for the left action, 0.6 for the straight action, and 0.4 for the right action, while the second agent's regularization prior  $\pi_{0,2} = [0.4, 0.6, 0.0]$  specified a probability of 0.4 for the left action, 0.6 for the straight action, and 0.0 for the right action. Clearly, the two agents have learned different strategies in the navigation problem. In Figure 3, it is clear that the two agents consistently took the shortest path to their respective destinations while adhering to their individual constraints. We therefore characterize them as agents that preferred to take right and left turns, respectively. Crucially, this is an *intrinsic* property of the agents imposed by the regularization of the objective function. This separates our method of intrinsic characterization from post hoc characterization techniques.

Finally, Figure 4 shows typical curves of training and testing returns for both the single-agent and multi-agent systems across 3000 training iterations. The agents clearly demonstrate a good learning response with steadily increasing returns both in training and testing and, while training performance is naturally slightly dependant on random initial conditions, there is no significant difference in convergence time between the single-agent and multi-agent systems.



**Figure 3.** Four sets of trajectories for a dual-agent environment in the navigation problem. The first agent—labelled ‘right turns’—received a regularization prior  $\pi_{0,1} = [0.0, 0.6, 0.4]$  while the second agent—labelled ‘left turns’—received a regularization prior  $\pi_{0,2} = [0.4, 0.6, 0.0]$ , where the values in  $\pi_{0,i}$  correspond to the probabilities of the actions turn left, go straight, and turn right. In (a) both agents’ destinations are on the left, but only the agent regularized to prefer left turns actually turns left while the other agent completes a series of right turns to reach its destination. In (b) both agents’ destinations are located such that they have to perform a series of turns according to their regularization priors. In (c) both agents’ destinations are located such that they perform their preferential turn—either to the left, or to the right—according to their regularization priors. In (d) one agent’s destination is straight ahead and it needs to not turn; the regularization prior clearly allows for such a strategy.



**Figure 4.** Training and testing returns for two typical training runs: the single-agent system in (a) and the multi-agent system in (b). In both cases, the learning processes clearly followed steady increases in returns and convergence happened roughly in the same number of iterations.

## 5. Conclusions and Direction for Future Work

Our objective was the *intrinsic* characterization of RL agents. To this end, we investigated and briefly summarized the relevant state-of-the-art in explainable RL and found that these methods have typically been relying on post-hoc evaluations of a learned policy. Policy regularization is a method that modifies a policy; however, it has typically been employed to enhance training performance which does not necessarily aid in policy characterization. We therefore adapted entropy regularization from maximizing the entropy in the policy to minimizing the mean squared difference between the expected action and a given prior. This encourages the agent to mimic a predefined behaviour while maximizing its reward during learning. Finally, we extended MADDPG with our regularization term. We provided a formal argument for the validity of our algorithm and empirically demonstrated its functioning in a toy problem. In this problem, we characterized two agents to follow different approaches when navigating to a destination in a grid; while one agent performed only right turns, the other performed only left turns. We conclude that our fundamentally sound algorithm was able to imbue our agents’ policies with specific characteristic behaviours. In future work, we intend to use this algorithm to develop a set of financial advisors that will optimize individual customers’ investment portfolios according to their individual spending personalities [24]. While maximising portfolio values, these



agents may prefer, e.g., property investments over crypto currencies, which are analogous to right turns and left turns in our toy problem.

**Author Contributions:** Conceptualization, C.M. and C.O.; methodology, C.M.; software, C.M.; validation, C.M.; formal analysis, C.M.; investigation, C.M.; resources, C.M.; data curation, C.M.; writing—original draft preparation, C.M.; writing—review and editing, C.O. and C.M.; visualization, C.M.; supervision, C.O.; project administration, C.M.; funding acquisition, C.M. and C.O.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by The Norwegian Research Foundation, project number 311465.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

## References

1. Heuillet, A.; Couthouis, F.; Díaz-Rodríguez, N. Explainability in deep reinforcement learning. *Knowl. Based Syst.* **2021**, *214*, 1–24. [[CrossRef](#)]
2. García, J.; Fernández, F. A Comprehensive Survey on Safe Reinforcement Learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480.
3. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [[CrossRef](#)]
4. Wells, L.; Bednarz, T. Explainable AI and Reinforcement Learning: A Systematic Review of Current Approaches and Trends. *Front. Artif. Intell.* **2021**, *4*, 1–48. [[CrossRef](#)] [[PubMed](#)]
5. Gupta, S.; Singal, G.; Garg, D. Deep Reinforcement Learning Techniques in Diversified Domains: A Survey. *Arch. Comput. Methods Eng.* **2021**, *28*, 4715–4754. [[CrossRef](#)]
6. Haarnoja, T.; Tang, H.; Abbeel, P.; Levine, S. Reinforcement Learning with Deep Energy-Based Policies. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia, 6–11 August 2017.
7. Galashov, A.; Jayakumar, S.; Hasenclever, L.; Tirumala, D.; Schwarz, J.; Desjardins, G.; Czarnecki, W.M.; Teh, Y.W.; Pascanu, R.; Heess, N. Information asymmetry in KL-regularized RL. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
8. Lu, J.; Dissanayake, S.; Castillo, N.; Williams, K. *Safety Evaluation of Right Turns Followed by U-Turns as an Alternative to Direct Left Turns—Conflict Analysis*; Technical Report, CUTR Research Reports 213, University of South Florida, Scholar Commons, Tampa, FL, USA, 2001.
9. Riveret, R.; Gao, Y.; Governatori, G.; Rotolo, A.; Pitt, J.V.; Sartor, G. A probabilistic argumentation framework for reinforcement learning agents. *Auton. Agents Multi-Agent Syst.* **2019**, *33*, 216–274. [[CrossRef](#)]
10. Madumal, P.; Miller, T.; Sonenberg, L.; Vetere, F. Explainable Reinforcement Learning Through a Causal Lens. *arXiv* **2019**, arXiv:1905.10958v2.
11. van Seijen, H.; Fatemi, M.; Romoff, J.; Laroché, R.; Barnes, T.; Tsang, J. Hybrid Reward Architecture for Reinforcement Learning. *arXiv* **2017**, arXiv:1706.04208.
12. Juozapaitis, Z.; Koul, A.; Fern, A.; Erwig, M.; Doshi-Velez, F. Explainable Reinforcement Learning via Reward Decomposition. In Proceedings of the International Joint Conference on Artificial Intelligence. A Workshop on Explainable Artificial Intelligence, Macao, China, 10–16 August 2019.
13. Beyret, B.; Shafti, A.; Faisal, A. Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5014–5019.
14. Marzari, L.; Pore, A.; Dall’Alba, D.; Aragon-Camarasa, G.; Farinelli, A.; Fiorini, P. Towards Hierarchical Task Decomposition using Deep Reinforcement Learning for Pick and Place Subtasks. *arXiv* **2021**, arXiv:2102.04022.
15. Sequeira, P.; Yeh, E.; Gervasio, M. Interestingness Elements for Explainable Reinforcement Learning through Introspection. *IIUI Work.* **2019**, *2327*, 1–7.
16. Littman, M.L. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In Proceedings of the Eleventh International Conference on Machine Learning, New Brunswick, NJ, USA, 10–13 July 1994; pp. 157–163.

17. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations (ICLR) (Poster), San Juan, Puerto Rico, 2–4 May 2016.
18. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS, Long Beach, CA, USA, 4–9 December 2017.
19. Ziebart, B.D. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. Ph.D. Thesis, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.
20. Cheng, R.; Verma, A.; Orosz, G.; Chaudhuri, S.; Yue, Y.; Burdick, J.W. Control Regularization for Reduced Variance Reinforcement Learning. *arXiv* **2019**, arXiv:1905.05380.
21. Parisi, S.; Tangkaratt, V.; Peters, J.; Khan, M.E. TD-regularized actor-critic methods. *Mach. Learn.* **2019**, *108*, 1467–1501. [[CrossRef](#)]
22. Miryoosefi, S.; Brantley, K.; Daume III, H.; Dudik, M.; Schapire, R.E. Reinforcement Learning with Convex Constraints. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 1–10.
23. Chow, Y.; Ghavamzadeh, M.; Janson, L.; Pavone, M. Risk-Constrained Reinforcement Learning with Percentile Risk Criteria. *J. Mach. Learn. Res.* **2015**, *18*, 1–51.
24. Maree, C.; Omlin, C.W. Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality (In Print). In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 4–7 December 2021; pp. 1–5.



## Appendix F

# Can Interpretable Reinforcement Learning Manage Prosperity Your Way?


This paper has been published as:

C. Maree and C. W. Omlin, “Can Interpretable Reinforcement Learning Manage Prosperity Your Way?”, *AI*, **2022**, 3(2), pp. 526–537, doi: 10.3390/ai3020030.

Copyright © Creative Commons Attribution (CC BY)

Article

# Can Interpretable Reinforcement Learning Manage Prosperity Your Way?

Charl Maree <sup>1,2,\*</sup>  and Christian W. Omlin <sup>2,†</sup><sup>1</sup> Chief Technology Office, Sparebank 1 SR-Bank, 4007 Stavanger, Norway<sup>2</sup> Center for AI Research, University of Agder, 4879 Grimstad, Norway; christian.omlin@uia.no

\* Correspondence: charl.maree@uia.no

† Current address: Jon Lilletunsvai 9, 4879 Grimstad, Norway.

**Abstract:** Personalisation of products and services is fast becoming the driver of success in banking and commerce. Machine learning holds the promise of gaining a deeper understanding of and tailoring to customers' needs and preferences. Whereas traditional solutions to financial decision problems frequently rely on model assumptions, reinforcement learning is able to exploit large amounts of data to improve customer modelling and decision-making in complex financial environments with fewer assumptions. Model explainability and interpretability present challenges from a regulatory perspective which demands transparency for acceptance; they also offer the opportunity for improved insight into and understanding of customers. Post-hoc approaches are typically used for explaining pretrained reinforcement learning models. Based on our previous modeling of customer spending behaviour, we adapt our recent reinforcement learning algorithm that intrinsically characterizes desirable behaviours and we transition to the problem of prosperity management. We train inherently interpretable reinforcement learning agents to give investment advice that is aligned with prototype financial personality traits which are combined to make a final recommendation. We observe that the trained agents' advice adheres to their intended characteristics, they learn the value of compound growth, and, without any explicit reference, the notion of risk as well as improved policy convergence.



**Citation:** Maree, C.; Omlin, C.W. Can Interpretable Reinforcement Learning Manage Prosperity Your Way? *AI* **2022**, *3*, 526–537. <https://doi.org/10.3390/ai3020030>

Academic Editors: José Manuel Ferreira Machado and Kenji Suzuki

Received: 6 May 2022  
Accepted: 10 June 2022  
Published: 13 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** AI in banking; personalized services; prosperity management; explainable AI; reinforcement learning; policy regularisation

## 1. Introduction

Personalization is critical in modern retail services, and banking is no exception. Financial service providers are employing ever-advancing methods to improve the level of personalisation of their services [1,2]. Artificial intelligence (AI) is a promising tool in this pursuit in areas such as anti-money laundering, trading and investment, and customer relationship management [3]. Examples of *personalised* services are recommender systems for product sales [4], risk evaluation for credit scoring [5], and segmentation for customer-centric marketing [6]. More commonly, AI has been applied to stock trading via ensemble learning [7], currency recognition using deep learning [8], stock index performance through time-series modelling with feature engineering [9], and investment portfolio management using reinforcement learning (RL) [10,11]. These applications generally lack the personalisation needed to enhance customer relations and support service delivery for growing customer bases. We address the issue of personalization by using an interpretable RL algorithm to manage a portfolio of various asset classes according to individual spending behaviour. Whereas the current literature is only concerned with portfolio optimization, our objective is a more holistic prosperity management service, which includes a more diverse portfolio of asset classes. Such a service might improve customer interaction through personalization, enhance trust through interpretability, and contribute to customer acquisition and retention.

The lack of explainability and interpretability has thus far hindered the wider adoption of machine learning, mainly due to model opacity; model understanding is essential in financial services [12–14]. We distinguish between explainability and interpretability: explainability refers to a symbolic representation of the knowledge a model has learned, while interpretability is necessary for reasoning about a model's predictions. We have previously investigated the *interpretability* of systems of multiple RL agents [15]: a regularisation term in the objective function instilled a desired agent behaviour *during* training. For our current purpose of prosperity management, we create prototypical RL agents which have intrinsic affinities for certain asset classes. They have characteristic asset allocation strategies which are easy to interpret. The asset allocation preference of a real customer is an amalgam of the strategies which can be realized by a composition of the prototypical RL agents. Unlike work reported in [16] that investigate Boolean composition of RL agents, our challenge is to determine a RL agent composition which is a reflection of real customer preferences. In this paper, we investigate the efficacy of linear compositions of prototypical RL agents which represent real customers; the coefficients in these linear compositions are the fuzzy memberships of customers' prototypical personality traits. We rate asset classes, such as stocks and savings accounts, in terms of their inherent properties, such as expected long term risk and reward, and liquidity. For each asset class property, we define an association with the prototypical personality traits [17]. We derive the agents' affinities for certain asset classes as the inner product of these associations and the asset class ratings. Their intrinsic interpretability may fulfill the promise of a digital private assistant for personal wealth management.

We introduce the relevant theoretical background in the next section, after which we discuss our methodology and list a set of key assumptions, present and discuss our results, and conclude with a discussion and suggestions for future work.

## 2. Related Work

Recent evidence has revealed a causal relationship between spending patterns and individual happiness [18]: we are happiest when our spending matches our personality. For instance, extraverted individuals typically prefer spending at a bar rather than at a bookshop, while the opposite may apply to introverts. Our premise is that spending personality traits can be carried over to prosperity management: we are happiest when our investment matches our personality. For instance, conscientious investors may prefer the predictability of property over the volatility of stocks. This is consistent with the high affinity of conscientious spenders towards residential mortgages [18]. It is compelling to expand the notion of personality traits from spending to wealth creation, i.e., to base personal investment advice on historical spending behaviour [19,20].

In RL, agents learn by trial and error to maximize the expected cumulative reward given by the environment in which they act [21]. Their actions result in changing the internal state of the environment, which is known to the agents through observations. RL agents are adept at maximising future rewards despite potential sparse or immediate negative rewards [21]. The environment is modelled as a Markov decision process (MDP), which is a discrete-time stochastic process in which the core underlying assumption is that the state of the environment depends solely on its previous state and the action taken by the agent [22]. It is described by the set  $(S, A, P, R)$  where  $S$  is a set of states,  $A$  a set of actions,  $P(s, a) = P(s_{t+1} = s' | s, a)$  the probability that action  $a$  in state  $s$  will lead to state  $s'$ , and  $R(s, a)$  is the reward given for action  $a$  in state  $s$ . Deep deterministic policy gradient (DDPG [23]) is a RL algorithm that represents an agent through two neural networks: an actor and a critic. The actor takes the state observation as input and predicts the best action, while the critic takes the state observation and predicted action as input and predicts the reward from the environment. While the critic learns the dynamics of the environment, the agent learns to maximize the predicted reward. For numerical stability and to improve convergence, DDPG initializes two identical target networks for the actor

and critic, respectively. The parameters of these target networks are slowly updated, as specified by the target update hyperparameter.

RL has been extensively applied to stock portfolio management [24–29], but not yet to holistic prosperity management; the lack of model transparency may be a contributing factor. Interpretation of RL agents typically follows model training [30–32]; our ambition is to impose a desired characteristic behaviour during training, thus making it an intrinsic property of the agent. Based on a prior that defines a desired behaviour, we extend the DDPG objective function with a regularisation term [15]. Formally, for each agent  $i$ , this objective function is given by:

$$J(\theta_i) = \mathbb{E}_{o_i, a_i \sim \mathcal{D}}[R_i(o_i, a_i)] - \lambda L_i \quad (1)$$

$$L_i = \frac{1}{M_i} \sum_{j=0}^{M_i} \left[ \mathbb{E}_{a \sim \pi_{\theta_i}}(a_j) - (a_j | \pi_{0_i}(a)) \right]^2$$

where  $\theta_i$  is a set of parameters governing the policy,  $\mathcal{D}$  is the replay buffer,  $R_i(o_i, a_i)$  is the reward for action  $a_i$  with the partial state observation  $o_i$ ,  $\lambda \in \mathbb{R}_{\geq 0}$  is a scaling parameter,  $M_i$  is the number of actions, and  $\pi_{0_i}$  is the prior that defines the desired behaviour of the agent. Note that the prior is independent of the state, which simplifies it and thus makes it interpretable; this is a departure from traditional policy regularisation methods such as KL-regularisation and entropy regularisation which aim to improve learning convergence instead [33,34]. Traditional regularisation encourages state space exploration by increasing the entropy of the policy, whereas our method guides agents' learning towards the prior and thus imposes a desired characteristic behaviour.

### 3. Methodology

The aim of this work was to create an interpretable AI for personal investment management. We used a policy regularisation method to instill inherent agent behaviours based on a prior action distribution, as in Equation (1), for which we detail the algorithm in Algorithm 1. Our underlying assumption is that our method finds a local optimum in close proximity to the regularisation prior, which we base on the fact that policy regularisation in general does not a-priori prevent the exploration-exploitation process from finding an optimum [33,34].

We selected five asset classes in which a customer could invest a monthly amount over a duration of 30 years: a savings account, property, a portfolio of stocks, luxury expenditures, and additional mortgage payments. We include luxury expenditure to the portfolio under the premise that it may increase customer satisfaction in their portfolios [18]. We define luxury items as any expenditure that may appeal to a person's personality profile; people scoring high on openness might derive joy from spending money on travelling, people scoring high on extraversion may prefer to spend money on festivities with other people [18], while other luxury items such as cars or artwork are also possible. While this investment class includes items typically listed on indices such as the Knight Frank luxury investment index [35]—art, fine wines, classic cars, etc., it also includes luxury expenditures such as travel, fine dining, and consumer electronics. However, it excludes basic household spending such as groceries, insurance, fuel, etc. Finally, we modelled the growth rates of assets according to historical index data, which we describe below.

**Algorithm 1** Policy regularisation algorithm from [15].

---

Initialize the actor  $\mu_{\theta_{\mu}}$  with random parameters  $\theta_{\mu}$   
Initialize the critic  $Q_{\theta_Q}$  with random parameters  $\theta_Q$   
Initialize the target actor  $\mu'_{\theta_{\mu'}}$  with parameters  $\theta_{\mu'} \leftarrow \theta_{\mu}$   
Initialize the target critic  $Q'_{\theta_{Q'}}$  with parameters  $\theta_{Q'} \leftarrow \theta_Q$   
Set the prior  $\pi_0$  and the number of actions  $M_i \leftarrow |\pi_0|$   
Set regularisation weight hyperparameter  $\lambda$   
Set target update rate hyperparameter  $\tau$   
Initialize the replay buffer  $\mathcal{D}$   
**for**  $e = 1$ , episodes **do**  
  Initialize a random exploration function  $F(e) \sim N(0, \sigma_e)$   
  Reset the environment and get the first state observation  $s_1$   
   $t \leftarrow 1$ ,  $Done \leftarrow False$   
  **while** not  $Done$  **do** ▷ Gather experience  
    Select the action and add exploration randomness  $a_t \leftarrow \mu_{\theta_{\mu}}(s_t) + F(e)$   
    Retrieve the environmental response: reward  $r_t$  and observation  $s'_t$   
    Store the transition tuple  $\mathcal{T} = (s_t, a_t, r_t, s'_t)$  to replay buffer:  $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{T}$   
     $t \leftarrow t + 1$   
     $s_t \leftarrow s'_t$   
    **if** (end of episode) **then**  
       $Done \leftarrow True$   
    **end if**  
  **end while**  
  Sample a random batch from the replay buffer  $\mathcal{B} \subset \mathcal{D}$  ▷ Learn using experience  
  replay  
   $\bar{Q} \leftarrow r_{\mathcal{B}} + \gamma Q'(s_{\mathcal{B}}, \mu'_{\mathcal{B}})$   
  Update critic parameters  $\theta_Q$  by minimising the loss:  

$$\mathcal{L}(\theta_Q) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} (Q_{\theta_Q} - \bar{Q})^2$$
  
  Update the actor parameters  $\theta_{\mu}$  by minimising the loss: ▷ From Equation (1)  

$$\mathcal{L}(\theta_{\mu}) = -\bar{Q} + \lambda \frac{1}{M} \sum_{j=1}^M [\bar{\mu}_j - (a_j | \pi_0)]^2$$
  
  Update the target parameters:  

$$\theta_{\mu'} \leftarrow \tau \theta_{\mu} + (1 - \tau) \theta_{\mu'}$$
  

$$\theta_{Q'} \leftarrow \tau \theta_Q + (1 - \tau) \theta_{Q'}$$
  
**end for**

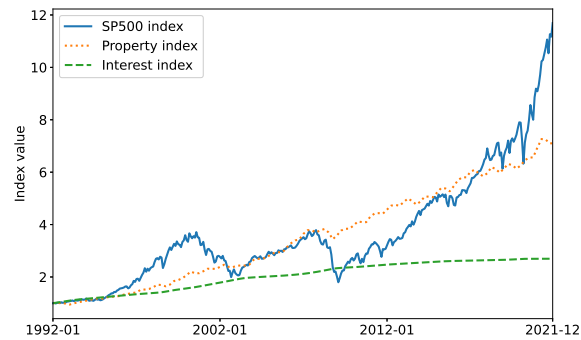
---

**3.1. Modelling Assumptions**

We continuously distribute funds into assets based on the indices of the S&P 500 [36], Norwegian property [37], and the Norwegian interest rate [38]. In addition, we invest in mortgages and luxury items. We show this data for a 30-year period in Figure 1.

We make a number of assumptions which limit the scope of the portfolio and simplify investment choices to make the characterization of agent behaviour and interpretation of investment strategies tractable.





**Figure 1.** Three asset value indices for a period of 30 years: The S&P 500 stock index, the Norwegian property index, and the Norwegian interest rate index. All indices are relative to their respective values on 1 January 1992. While the stock index performs the best overall, it has the highest volatility and therefore the highest risk. Conversely, the interest rate index has the lowest risk but also the lowest growth.

**Assumption 1.** Asset growth rates can be modelled by their respective asset indices, i.e., a stock portfolio may be modeled by a major stock index—e.g., the S&P 500, and an investment in property by its corresponding index.

The outright investment in indices such as S&P 500 is very common; it will return the growth rates according to these indices. This is a conservative assumption as stock portfolio optimization frequently outperforms indices, which may serve as a performance measure of the investment strategy [29].

To give personalised advice, we depart from the premise that there is a mere correlation between spending behaviour and happiness. We are expanding the notion of the causal relationship of spending patterns and customer satisfaction to chart an investment strategy and provide advice that is aligned with customer personality [18]. We enlisted a panel of experts from a major Norwegian bank to rate our asset classes according to a set of inherent properties: expected long term risk and returns, liquidity, minimum investment limits, and perceived novelty. We used the Sharpe ratio—the difference between the expected daily return and risk-free return divided by the standard deviation of daily returns—to quantify risk and historical data to gauge expected returns. These coefficients, the elements of a matrix  $P$ , are shown in Table 1.

**Table 1.** A matrix  $P$  rating the performance of each asset class with respect to a set of desirable properties. Values are in range  $[0, 1]$  and represent a relative low to high score in each of the properties.

Asset Class Property	Savings	Property	Stocks	Luxury	Mortgage
High expected long term returns	0.25	0.67	1.00	0.05	0.50
Low expected long term risk	1.00	0.32	0.10	0.05	1.00
High asset liquidity	1.00	0.25	0.80	0.10	0.05
Low minimum investment	0.80	0.25	1.00	0.50	1.00
High perceived novelty	0.10	0.25	0.75	1.00	0.10

The same panel of experts also assigned a matrix  $Q$  describing the likely associations between the prototypical personality traits and the asset classes, shown in Table 2. For instance, the conscientiousness trait might prefer assets classes with low expected risk, while the openness trait might prefer those which they perceive as novel.

**Table 2.** A matrix  $Q$  describing the association between prototypical personality traits—openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N)—and a set of inherent properties of each asset class. Values are in  $\{n \in \mathbb{Z} \mid -2 \leq n \leq 2\}$  and represent a highly negative, negative, neutral, positive and highly positive association, respectively.

Asset Class Property	O	C	E	A	N
High expected long term returns	1	1	2	1	1
Low expected long term risk	−1	2	−1	1	2
High asset liquidity	2	−1	2	1	2
Low minimum investment	0	−1	1	1	1
High perceived novelty	2	0	2	0	−1

From  $P$  and  $Q$ , we calculated a set of coefficients that describe the association that each personality trait might have with each of the asset classes. The resulting matrix of coefficients  $R = (Q^T \cdot P^T)^T$ , normalized by column and scaled such that the values are in the range  $[-1, 1]$ , are shown in Table 3.

**Table 3.** Coefficients relating asset risk, expected return, liquidity, capital requirement, and novelty to prototypical personality traits: openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N). The values are in the range  $[-1, 1]$ .

Investment	O	C	E	A	N
Savings	−0.11	0.08	−0.15	0.51	0.68
Property	−0.15	0.32	−0.22	−0.36	−0.24
Stocks	0.82	−0.61	0.95	0.42	0.12
Luxury	0.16	−0.51	−0.07	−0.80	−0.81
Mortgage	−0.72	0.72	−0.52	0.23	0.25

We define a MDP for a multi-agent RL setting as follows:

**States** A set of 13 continuous values representing the customer age (between 30 and 60 years and normalized to a range of  $[0, 1]$ ), six values for the asset class holdings and total portfolio value (scaled by  $1 : 10^6$ ), and two market indicators for each of the three indices, i.e., their mean asset convergence divergence (MACD) (the difference between the 26-month and the 12-month exponential moving average of a trend) which predicts trend reversals and relative strength index ( $RSI = 100 - 100 / (1 + \frac{P_x}{N_x})$ ) where  $P_x$  and  $N_x$  are the average positive and negative changes to the index values respectively, for  $x$  periods) which corrects for potential false predictions by MACD. The time horizon is 30 years.

**Reward** The changes in portfolio values between time steps.

**Actions** The continuous distribution of funds across the five asset classes.

**Assumption 2.** The initial values for a portfolio consist of a mortgage of NOK 2 million and a property valued at NOK 2 million. All other assets have zero initial value.

It is easy to adjust these initial portfolio assignments for different individuals.

**Assumption 3.** We make consistent monthly investments of 10,000 Norwegian kroner (NOK).

This can be easily modified for individual customers' contributions.

There is a priori no lower limit on the investment amounts:

**Assumption 4.** Property investment does not require bulk payments, i.e., smaller investments can be made through property funds, trusts, or crowdfunding.

While investment in physical real estate normally requires larger deposits, we allow our agents to invest smaller amounts into the property market, i.e., a fraction of the monthly investment contribution specified in Assumption 3. This is not a strong assumption as it is possible to invest smaller amounts in property indices, trusts, funds, etc.

We assign interest rates for savings accounts at 5–10% below, and those of mortgage accounts at 5–10% over the interest index. Individuals younger than 35 years receive the more beneficial interest rate, as is common in Norwegian banks. Luxury items experience a depreciation of 20% per year; the depreciation of luxury items is highly variable and depends on the item, e.g., while artwork may appreciate, cars typically depreciate rapidly:

**Assumption 5.** *Luxury items depreciate at 20% per year.*

Dividends are normally included in the calculation of indices and monthly transactions are relatively infrequent compared to high frequency trading:

**Assumption 6.** *Any additional income from investments—such as dividend payouts or rental income—as well as costs such as transaction costs and fund management costs are ignored.*

### 3.2. Agents

We train five DDPG agents, one for each of the five personality traits. Using Equation (1) we regularise their objective functions with a prior derived from their respective personality traits in Table 3, e.g., the openness prior  $\pi_0^O$  places the most weight on stocks and avoids mortgage repayments, property investment, and savings, while the conscientiousness prior  $\pi_0^C$  places the most weight on mortgage repayments and avoids stocks and luxury expenditure. These priors, shown in Table 4, are probability distributions across the investment channels and therefore add up to one.

**Table 4.** Regularisation priors  $\pi_0^a$  for each agent  $a \in \{\text{openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N)}\}$ .

Investment	$\pi_0^O$	$\pi_0^C$	$\pi_0^E$	$\pi_0^A$	$\pi_0^N$
Savings	0.00	0.07	0.00	0.44	0.64
Property	0.00	0.28	0.00	0.00	0.00
Stocks	0.84	0.00	1.00	0.36	0.12
Luxury	0.16	0.00	0.00	0.00	0.00
Mortgage	0.00	0.65	0.00	0.02	0.24

Our five agents have identical actor and critic networks, respectively. This is appropriate because they solve the same problem, but aim to find locally optimum policies in specific regions of the state-action space, as given by their respective regularisation priors. The 10 neural networks for the agents' actors and critics each consist of two fully connected feed-forward layers with 2000 nodes in each layer. The actor networks each have a final soft-max activation layer while the critic networks have no final activations. The reason for the actors' softmax activation is to ensure the values for the actions add up to one, while the critics need no activation as the rewards need not be scaled. We tuned the hyperparameters using a one-at-a-time parameter sweep resulting in learning rates of 0.004 and 0.001 for the actors and critics respectively, target network update parameters of  $\tau = 0.05$ , and regularisation coefficients of  $\lambda = 2$ . Training batch sizes were 256 time steps and we sized the replay buffer to hold 2048 transitions. Each iteration collected 256 time steps and completed two training batches.

## 4. Results

Each of our investment agents learns an optimal investment strategy for their respective prototypical personality traits, for instance, openness. The final portfolio values after

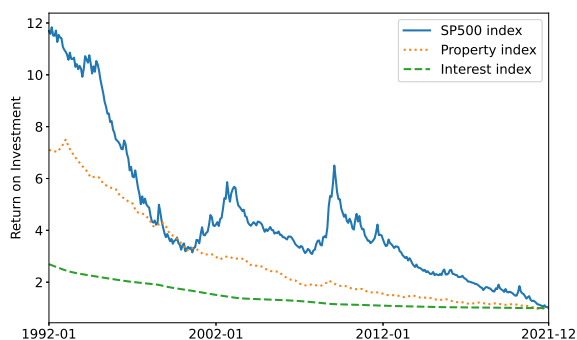
334 months of investing according to these policies are shown in Table 5. Given the common total investment of 3.34 million NOK, the compound annual growth rate varies between 5.8% and 7.8% which is the maximum return possible if investing in stocks only.

**Table 5.** Portfolio values of the five optimal policies for each of the prototypical personality traits.

Policy	Final Portfolio Value (NOK 1M)
Openness	22.4
Conscientiousness	18.8
Extraversion *	27.7
Agreeableness	20.5
Neuroticism	16.4
Personal agent	20.3

\* This agent's regularisation prior was coincidentally the same as the optimal monetary policy  $\pi^M$  and it achieved the maximum possible final portfolio value.

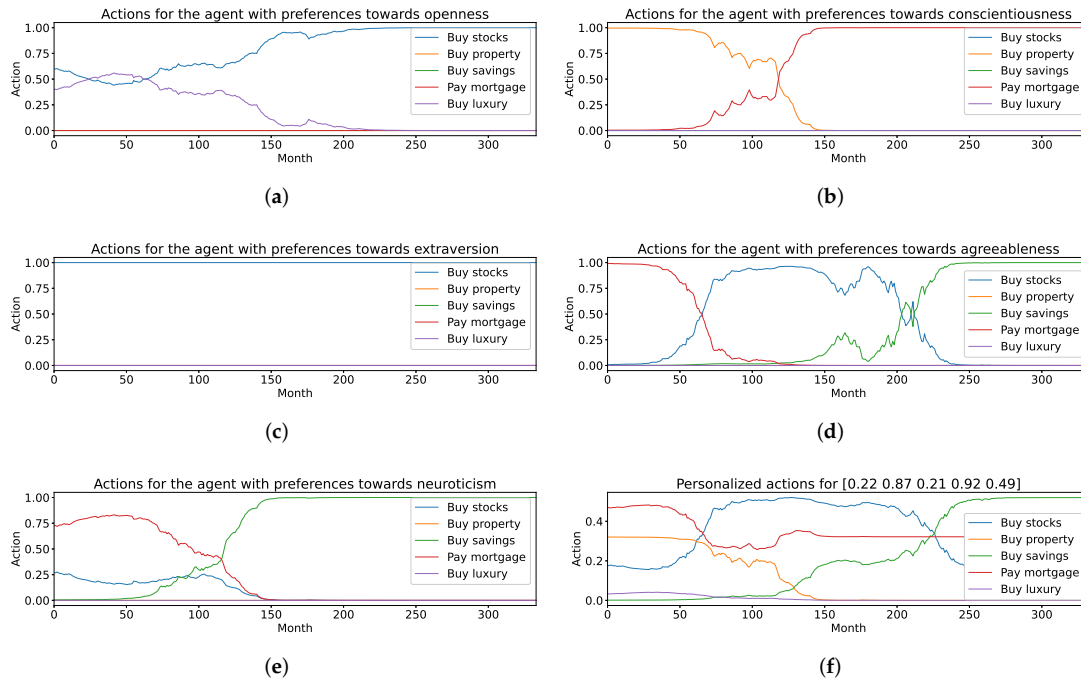
Note that these personalised policies did not achieve the same final portfolio value. In fact, the optimum policy in monetary terms  $\pi^M$  in this case would have been to always buy stocks as shown in Figure 2; this is the default policy an agent will converge towards when personality traits are ignored.



**Figure 2.** The return on investment at every time step, calculated as the index value at the final time step divided by the index value at the current time step. It is clear that S&P 500 has the greatest return on investment at every time step, except for a brief period in ca. 2000 where it was marginally below the property index. Therefore, the optimum monetary policy  $\pi^M$  is to always invest the maximum amount into stocks.

However, we postulate that this is not the ideal personal financial advice to give to all individuals; some customers may be more averse to risk and will thus prefer to avoid volatility in their portfolio. Our personalized agent takes into account such preferences and, e.g., it recommends property investments rather than stock investments.

Thus far, our agents have each separately learned an optimal investment strategy for each prototypical personality trait. The aggregate policy is the weighted sum of these individually learned policies: a customer has a blend of personality traits which can be represented as a vector with five entries with values within the range  $[-1, +1]$ . We calculate the inner product of the normalized personality vector and the prototypical policies to arrive at the aggregate investment policy. We show a representative aggregate investment policy for a customer with a random personality profile in Figure 3.



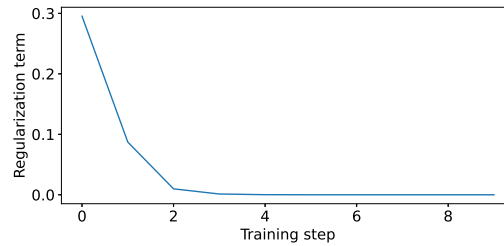
**Figure 3.** Investment strategies for different prototypical personality traits: (a–e) show the fractions of monthly investments for different assets. They reveal the distinct investment strategies with changing asset preferences for the five prototypical personality traits. In (f) we illustrate the investment strategy for a fictitious customer with a random personality profile [openness, conscientiousness, agreeableness, extraversion, and neuroticism] = [0.22, 0.87, 0.21, 0.92, 0.49]. The customer invests in a mixture of assets throughout the investment period.

We observe that the openness agent is the only agent to recommend spending on luxury items; this is to be expected because its regularisation prior  $\pi_0^O$  is the only one with a non-zero coefficient for luxury purchases. We also observe that the conscientiousness agent recommends investing in property in early stages, followed by rigorous loan repayments in the second half of the investment period. This suggests that our agent has learned the concept of compound growth and its utility for portfolio optimization. By contrast, the extraversion agent was steadfast in purchasing stocks only, which is consistent with its regularisation prior  $\pi_0^E$ . Unlike the conscientiousness agent, the agreeableness and neuroticism agents consistently recommend investing in savings towards the end of the investment period. In the early stages of the investment period, the agreeableness and neuroticism agents utilize compound growth to increase the portfolio value; in the latter phases, their regimen changes and they prefer the safety of savings accounts. This is noteworthy because although risk is not explicitly part of either the reward or regularisation functions, it is consistent with traditional financial advice, which decreases the risk level with age. Repeated training produces consistent results. We intend to elucidate this observation in future work.

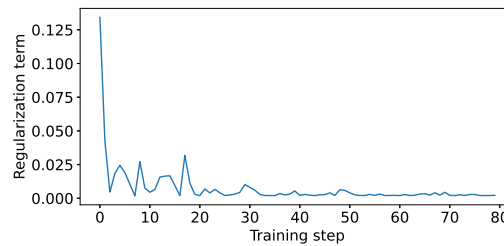
We observe that training converges quickly to the desired behaviour (see Figure 4); the contribution of the regularisation term decreases rapidly, which implies that the agent is learning the intended behaviour. We show the regularisation term for the extraversion agent where the regularisation prior  $\pi_0^E$  matches the optimum monetary policy  $\pi^M$  in Figure 4a. Further training causes no instability as is often observed in the DDPG algorithm [34]. We

hypothesize that this may be due to the agent characteristics imposed by our regularisation whose effect may be similar to entropy regularisation [34].

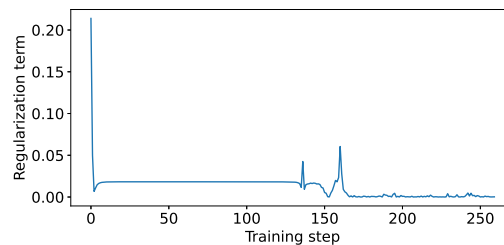
The actions of any linear combination of these agents, i.e., any personal agent, are interpretable through the intrinsic characterizations, i.e., priors, of each of the regularized agents.



(a)



(b)



(c)

**Figure 4.** The regularisation term  $L$  for three different runs. In (a) the regularisation prior  $\pi_0^E$  of the extraverted agent coincides with the optimum monetary policy  $\pi^M$  and the policy converges within 5 time steps. (b) shows a typical training run for the other agents which converges within 100–200 training steps. (c) shows a training run where the regularisation term appears to fall in local minimum for a time, but eventually finds the optimum after about 200 training steps.

## 5. Conclusions and Directions for Future Work

We have presented a novel application of training RL agents to exhibit desired characteristics and behaviours in prosperity management. The method is based on the regularisation of the policy *during* training. Here, we use prototypical personality traits—openness, conscientiousness, agreeableness, extraversion, and neuroticism—to define a set of priors which express their affinity towards different assets and thus impose different investment strategies. This makes the agents' behaviour explicit and thus offers an explanation for their recommendations. Our agents learn distinct optimal strategies for the continuous distribution of monthly investments across a portfolio of investment assets. We have shown

that the agents learned to optimize total rewards while adhering to their distinct priors. This makes it possible to interpret the agents' investment strategies.

Unlike traditional DDPG algorithms which may diverge with continuous training, our regularisation results in quick and robust convergence. This could become relevant if RL agents undergo continuous training to give personalized investment advice to customers. The justification of this observation will be subject to future research. Further, our regularisation method encourages exploration of a specific region in the action space, defined by the prior  $\pi_0$ , which leads to a local optimum in near proximity of the prior. This is a specific case of the generalised entropy regularisation, which expedites convergence to the global optimum policy by encouraging exploration of the entire state-action space.

Our agents have learned the concept and utility of compound growth rates and risk avoidance, which form part of the interpretation of their investment strategies. These are solely based on the regularisation priors which express their personality traits; the reward function makes no reference to the personality traits. While the notion of compound growth may emerge from the reward function, we do not yet know whether the notion of risk avoidance is connected to the reward function or regularisation.

Here, we have chosen a linear combination of different, separately trained agents aligned with the prototypical personality traits to arrive at an aggregate investment advice. In the future, we will investigate whether the orchestration of these agents can be learned to approach the optimum monetary policy. This aggregation will need an explanation as well as interpretation to understand its impact on the investment strategy. The hierarchical orchestration of prototypical agents will be learned from real customers' personality profiles. This will result in an explainable and interpretable personalized financial investment advisor.

**Author Contributions:** Conceptualization, C.M. and C.W.O.; methodology, C.M.; software, C.M.; validation, C.M.; formal analysis, C.M. and C.W.O.; investigation, C.M.; resources, C.M.; data curation, C.M.; writing—original draft preparation, C.M.; writing—review and editing, C.W.O. and C.M.; visualization, C.M.; supervision, C.W.O.; project administration, C.M. and C.W.O.; funding acquisition, C.M. and C.W.O.; All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially funded by The Norwegian Research Foundation, project number 311465.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study, in the collection, analyses, or interpretation of data, in the writing of the manuscript, or in the decision to publish the results.

## References

1. Stefanel, M.; Goyal, U. *Artificial Intelligence & Financial Services: Cutting through the Noise*; Technical Report; APIS Partners: London, UK, 2019.
2. Jaiwant, S.V. Artificial Intelligence and Personalized Banking. In *Handbook of Research on Innovative Management Using AI in Industry 5.0*; Vikas, G., Goel, R., Eds.; IGI Global: Bengaluru, India, 2022; pp. 74–87.
3. van der Burgt, J. *General Principles for the Use of Artificial Intelligence in the Financial Sector*; Technical Report; De Nederlandsche Bank: Amsterdam, The Netherlands, 2019.
4. Oyeboode, O.; Orji, R. A hybrid recommender system for product sales in a banking environment. *J. Bank. Financ. Technol.* **2020**, *4*, 15–25. [[CrossRef](#)]
5. Bhatore, S.; Mohan, L.; Reddy, R. Machine learning techniques for credit risk evaluation: A systematic literature review. *J. Bank. Financ. Technol.* **2020**, *4*, 111–138. [[CrossRef](#)]
6. Desai, D. Hyper-Personalization: An AI-Enabled Personalization for Customer-Centric Marketing. In *Adoption and Implementation of AI in Customer Relationship Management*; Singh, S., Ed.; IGI Global: Maharashtra, India, 2022; pp. 40–53.
7. Jothimani, D.; Yadav, S. Stock trading decisions using ensemble-based forecasting models: A study of the Indian stock market. *J. Bank. Financ. Technol.* **2019**, *3*, 113–129. [[CrossRef](#)]

8. Zhang, Q.; Yan, W.; Kankanhalli, M. Overview of currency recognition using deep learning. *J. Bank. Financ. Technol.* **2019**, *3*, 59–69. [[CrossRef](#)]
9. Hsu, T.Y. Machine learning applied to stock index performance enhancement. *J. Bank. Financ. Technol.* **2021**, *5*, 21–33. [[CrossRef](#)]
10. Kolm, P.; Ritter, G. Modern Perspectives on Reinforcement Learning in Finance. *SSRN Electron. J.* **2019**, *1*, 1–28. [[CrossRef](#)]
11. Fischer, T.G. *Reinforcement Learning in Financial Markets—A Survey*; Technical Report; Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics: Erlangen, Germany, 2018.
12. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [[CrossRef](#)]
13. Cao, L. AI in Finance: Challenges, Techniques and Opportunities. *Bank. Insur. eJournal* **2021**, *14*, 1–40. [[CrossRef](#)]
14. Maree, C.; Modal, J.E.; Omlin, C.W. Towards Responsible AI for Financial Transactions. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, Australia, 1–4 December 2020; pp. 16–21.
15. Maree, C.; Omlin, C. Reinforcement Learning Your Way: Agent Characterization through Policy Regularization. *AI* **2022**, *3*, 250–259. [[CrossRef](#)]
16. Tasse, G.N.; James, S.; Rosman, B. A Boolean Task Algebra for Reinforcement Learning. In Proceedings of the Neural Information Processing Systems, Online, 6–12 December 2020; Volume 34, pp. 1–11.
17. Gladstone, J.; Matz, S.; Lemaire, A. Can Psychological Traits Be Inferred From Spending? Evidence From Transaction Data. *Psychol. Sci.* **2019**, *30*, 1087–1096. [[CrossRef](#)] [[PubMed](#)]
18. Matz, S.C.; Gladstone, J.J.; Stillwell, D. Money Buys Happiness When Spending Fits Our Personality. *Psychol. Sci.* **2016**, *27*, 715–725. [[CrossRef](#)] [[PubMed](#)]
19. Maree, C.; Omlin, C.W. Clustering in Recurrent Neural Networks for Micro-Segmentation using Spending Personality. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–5.
20. Maree, C.; Omlin, C.W. Understanding Spending Behavior: Recurrent Neural Network Explanation and Interpretation. In Proceedings of the IEEE Computational Intelligence for Financial Engineering and Economics, Helsinki, Finland, 4–5 May 2022; pp. 1–7, *in print*.
21. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018.
22. Bellman, R. A Markovian decision process. *J. Math. Mech.* **1957**, *6*, 679–684. [[CrossRef](#)]
23. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971.
24. Bartram, S.M.; Branke, J.; Rossi, G.D.; Motahari, M. Machine Learning for Active Portfolio Management. *J. Financ. Data Sci.* **2021**, *3*, 9–30. [[CrossRef](#)]
25. Jurczenko, E. *Machine Learning for Asset Management: New Developments and Financial Applications*; Wiley-ISTE: London, UK, 2020; pp. 1–460.
26. Lim, Q.; Cao, Q.; Quek, C. Dynamic portfolio rebalancing through reinforcement learning. *Neural Comput. Appl.* **2021**, *33*, 1–15. [[CrossRef](#)]
27. Pinelis, M.; Ruppert, D. Machine learning portfolio allocation. *J. Financ. Data Sci.* **2022**, *8*, 35–54. [[CrossRef](#)]
28. Millea, A. Deep reinforcement learning for trading—A critical survey. *Data* **2021**, *6*, 119. [[CrossRef](#)]
29. Maree, C.; Omlin, C.W. Balancing Profit, Risk, and Sustainability for Portfolio Management. In Proceedings of the IEEE Computational Intelligence for Financial Engineering and Economics, Helsinki, Finland, 4–5 May 2022; pp. 1–8, *in print*.
30. Heuillet, A.; Couthouis, F.; Díaz-Rodríguez, N. Explainability in deep reinforcement learning. *Knowl.-Based Syst.* **2021**, *214*, 106685. [[CrossRef](#)]
31. Wells, L.; Bednarsz, T. Explainable AI and Reinforcement Learning: A Systematic Review of Current Approaches and Trends. *Front. Artif. Intell.* **2021**, *4*, 550030. [[CrossRef](#)] [[PubMed](#)]
32. Gupta, S.; Singal, G.; Garg, D. Deep Reinforcement Learning Techniques in Diversified Domains: A Survey. *Arch. Comput. Methods Eng.* **2021**, *28*, 4715–4754. [[CrossRef](#)]
33. Ziebart, B.D. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy. Ph.D. Thesis, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA, 2010.
34. Haarnoja, T.; Tang, H.; Abbeel, P.; Levine, S. Reinforcement Learning with Deep Energy-Based Policies. In Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017.
35. Knight Frank Company. Knight Frank Luxury Investment Index. 2022. Available online: <https://www.knightfrank.com/wealthreport/luxury-investment-trends-predictions/> (accessed on 27 May 2022).
36. Yahoo Finance. Historical Data for S&P500 Stock Index. 2022. Available online: <https://finance.yahoo.com/quote/> (accessed on 30 January 2022).
37. Statistics Norway. Table 07221—Price Index for Existing Dwellings. 2022. Available online: <https://www.ssb.no/en/statbank/table/07221/> (accessed on 30 January 2022).
38. Norges Bank. Interest Rates. 2022. Available online: <https://app.norges-bank.no/query/#/en/interest> (accessed on 30 January 2022).





# Appendix G

## Reinforcement Learning with Intrinsic Affinity for Personalized Prosperity Management

This paper has been published as:

C. Maree and C. W. Omlin, “Reinforcement Learning with Intrinsic Affinity for Personalized Prosperity Management”, *Digital Finance*, **2022**, 4(3), pp. 241–262, doi: 10.1007/s42521-022-00068-4.

Copyright © 2022 Springer Nature



# Reinforcement learning with intrinsic affinity for personalized prosperity management

Charl Maree<sup>1,2</sup>  · Christian W. Omlin<sup>1</sup>

Received: 19 April 2022 / Accepted: 30 August 2022  
© The Author(s) 2022

## Abstract

The purpose of applying reinforcement learning (RL) to portfolio management is commonly the maximization of profit. The extrinsic reward function used to learn an optimal strategy typically does not take into account any other preferences or constraints. We have developed a regularization method that ensures that strategies have global intrinsic affinities, i.e., different personalities may have preferences for certain asset classes which may change over time. We capitalize on these intrinsic policy affinities to make our RL model inherently interpretable. We demonstrate how RL agents can be trained to orchestrate such individual policies for particular personality profiles and still achieve high returns.

**Keywords** AI in banking · Personalized financial services · Explainable AI · Reinforcement learning · Policy regularization · Intrinsic affinity · Robo-advising

**JEL Classification** C10 · C30 · C32 · C40 · C45 · C50 · C51 · C52 · C53 · C54 · C58 · D10 · D14 · D31 · D53 · D91 · E22 · E37 · G11 · G41

## 1 Introduction

Effective customer engagement is a prerequisite for modern financial service providers that are adopting advanced methods to increase the level of personalization of their services (Stefanel & Goyal, 2019). Although artificial intelligence (AI) has become a ubiquitous tool in financial technology (Fernández, 2019), research in the field has yet to significantly advance levels of personalization (Maree & Omlin,

---

✉ Charl Maree  
charl.maree@uia.no

Christian W. Omlin  
christian.omlin@uia.no

<sup>1</sup> Center for Artificial Intelligence Research, University of Agder, Grimstad, Norway

<sup>2</sup> Chief Technology Office, Sparebank 1 SR-Bank, Stavanger, Norway

2021). Asset management is an active research topic in AI for finance; however, research opportunities presented by the need for personalized services are usually neglected (Millea, 2021). Whereas personalized investment advice is typically based on questionnaires, we propose a customer profiling from micro-segmentation that is based on spending behavior. Traditionally, customer segmentation has been grounded in demographics that provide only a coarse segmentation (Smith, 1956); it fails to capture nuanced differences between individuals with the potential for undesirable ramifications, e.g. discrimination in credit scoring based on postal code (Barocas & Selbst, 2016). Micro-segmentation, however, provides a more sophisticated classification that can improve the quality of banking services (Mousaeirad, 2020; Apeh et al., 2011).

We develop a personal prosperity manager that invests in a portfolio of asset classes according to individual personality profiles, as manifested by their spending behavior. The result is a hierarchical system of reinforcement learning (RL) agents in which a high-level agent orchestrates the actions of five low-level agents with global intrinsic affinities for certain asset classes. These affinities derive from prototypical personality traits. For instance, personality traits with a higher affinity for risk may, as a general rule, prefer high-volatility asset types.

Explainability and interpretability form the basis for understanding and trust (Barredo Arrieta et al., 2020). They are imperative for critical industries such as finance, but they have not yet been adequately addressed (Ramon et al., 2021; Cao, 2021). We regularize our agents' policies by predefined prior action distributions, thus imprinting characteristic behaviors that make their policies inherently interpretable on three levels: (1) the salient features extracted from customer spending behavior, (2) the affinities of the prototypical agents, and (3) their orchestration to achieve personal investment advice. Our contribution is, therefore, twofold: we demonstrate how RL agents can be made inherently interpretable through their intrinsic affinities, and we exemplify their value through their application in personalized prosperity management.

## 2 Background and related work

Recurrent neural networks (RNNs) are a class of artificial deep neural networks that are adept at processing temporal inputs. Their nodes maintain a memory of past events and learn representations in the form of activations (Hochreiter & Schmidhuber, 1997). It is established practice to investigate these node activations using the tools provided by the theory of dynamical systems (Ceni et al., 2019; Maheswaranathan et al., 2019). The state space of a RNN refers to the  $N$ -dimensional representation of the node activations, where  $N$  is the number of nodes in the RNN. For three (or fewer) nodes, their activation can, for example, be visualized in a three (or lower) dimensional plot, where each axis represents one node. This state-space plot is a useful implement for investigating the dynamics that govern the RNN. The theory of dynamical systems introduces the concept of attractors (Milnor, 2004); they are a set of states, or points in the state space, toward which neighboring states converge.

There are two main classes of attractors: fixed attractors, e.g., points, lines, surfaces, or other geometric shapes, and strange attractors that cannot be described as combinations of these, e.g., oscillating, chaotic, etc.

Gladstone et al. (2019) found that spending patterns are a predictor of financial personality. They trained a random forest to predict customer personalities from their classified financial transactions, using a prevalent taxonomy of personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. Although they achieved only a modest predictive accuracy, Tovanich et al. (2021) found that spending patterns over time expose salient information that is obscured in non-temporal form; the authors in this study used the same personality model, but added temporal patterns such as variability of the amount, persistence of the category in time, and burstiness—the intermittent changes in frequency of an event. Recurrent neural networks are able to extract this salient information when predicting personality traits from financial transactions (Maree & Omlin, 2021). In Maree and Omlin (2022c), we gained an understanding of these extracted features by interpreting the dynamics of the RNN state space through locating the set of attractors that govern the model. Understanding model behavior is crucial in industries such as personal finance (Ramon et al., 2021). In their study, Ramon et al. (2021) extracted rules from three classes of models—linear regression, logistic regression, and random forests—which not only exposed the spending patterns most indicative of personality traits, but also aided in model improvement.

In RL, agents learn to solve problems by tentation; they maximize the expected rewards resulting from their actions in an environment (Sutton & Barto, 2018). The expected reward  $R$  is the sum of discounted rewards for a time horizon controlled by a discount factor  $\gamma$ :  $R = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots + \gamma^T r_{t+T}$ . The environment is modelled as a Markov decision process (MDP) with sets of states  $S$ , actions  $A$ , rewards  $R(s, a)$ ,  $s \in S, a \in A$ , and transition probabilities  $P(s'|s, a)$ . Deterministic policy gradients (DDPG, Lillicrap et al., 2019) is an algorithm that maximizes the expected reward by learning a state-action value function  $Q(s, a)$  and the optimum action for each state  $\mu(s)$ . For numerical stability, it uses duplicate ‘target’ models  $Q'(s, a)$  and  $\mu'(s)$  for which the parameters  $\theta'$  are updated slowly using the soft-update formula:  $\theta' = \tau\theta + (1 - \tau)\theta'$  where  $\tau$  is normally a small value and  $\theta$  and  $\theta'$  refer to the main and target network parameters, respectively. Environments can have complex dynamics that result in sophisticated policies that are opaque to their developers, who may neither understand nor be able to control what these agents learn (Heuillet et al., 2021; García & Fernández, 2015). Intrinsic motivation enables agents to learn behaviors that are detached from the expected rewards of the environment (Aubret et al., 2019). It is a strategy that was developed to address the challenge of exploration in environments with sparse rewards (Andres et al., 2022). One such approach is Kullback-Leibler (KL) policy regularization in which the objective function is regularized by the KL-divergence between the current policy and a predefined prior (Galashov et al., 2019). Policy regularization has been shown to be helpful and never detrimental to convergence (Vieillard et al., 2022). Although most policy regularization methods aim to improve learning performance, they can also control the learning

process and imbue the policy with an intrinsic behavior (Maree & Omlin, 2022a). Here, the DDPG objective function is regularized with a predefined prior action distribution that defines a desirable characteristic:

$$J(\theta) = \mathbb{E}_{s,a \sim \mathcal{D}}[R(s, a)] - \lambda L$$

$$L = \frac{1}{M} \sum_{j=0}^M \left[ \mathbb{E}_{a \sim \pi_\theta} [a_j] - (a_j | \pi_0(a)) \right]^2 \quad (1)$$

$J(\theta)$  is the learning objective as a function of the model parameters  $\theta$ ,  $R(s, a)$  is the expected reward for state  $s$  and action  $a$  as sampled from a replay buffer  $\mathcal{D}$ , and  $\lambda$  is a scaling hyperparameter for the regularization term  $L$ , which is the mean square difference across  $M$  number of actions between the current action distribution and the action distribution given a regularization prior  $\pi_0$ . The efficacy of this approach was demonstrated by instilling an inherent characteristic behavior in agents that navigate a grid. These agents learned to either prefer left turns, right turns, or to avoid going straight by taking a zig-zag approach to their destination. In contrast to constrained RL which *avoids* certain states (Miryoosefi et al., 2019), the policy regularization in Maree and Omlin (2022a) *encourages* certain actions irrespective of the state and is a new direction for RL.

Hierarchical reinforcement learning (HRL) decomposes problems into low-level subtasks that are learned by relatively simple agents for the purpose of either improved performance or explainability (Pateria et al., 2021; Levy et al., 2019). Larger problems are solved by choreographing these subtasks through an orchestration agent that learns the high-level dynamics of its environment (Hengst, 2010). To our knowledge, there have been no applications of HRL in finance, and our work is the first. HRL has, however, been used to control a robotic arm: while low-level agents learned simple tasks such as moving forward/backward or picking up/placing down, an orchestration agent learned to retrieve objects on a surface by choreographing these tasks (Marzari et al., 2021; Beyret et al., 2019). The agents were not only efficient at learning, but their policies were more easily interpreted by human experts. Kulkarni et al. (2016) used HRL to train a hierarchical set of agents to play a game. Their low-level agents learned to solve simple tasks such as “pick up a key” or “open a door” while receiving extrinsic rewards from the environment. A high-level agent then orchestrated these sub-tasks and received intrinsic rewards generated by a critic based on whether or not larger objectives were met.

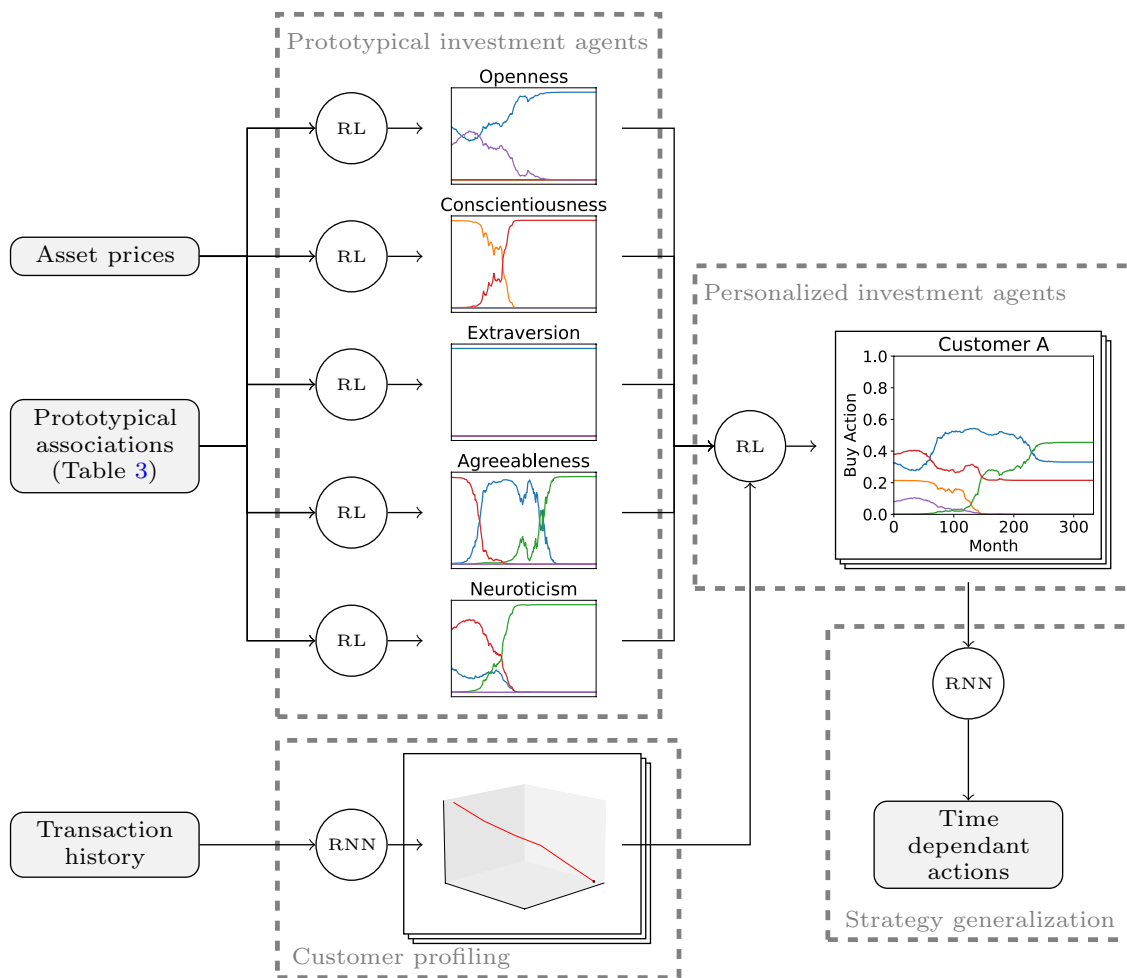
### 3 Methodology

To facilitate a comprehensive understanding of our work, we give a brief summary of previous work in learning prototypical investment strategies and customer profiling based on spending behavior. We discuss how we trained five low-level RL agents to invest in a set of asset classes according to prototypical personality traits (Maree & Omlin, 2022b), and how we extracted spending-behavioral trajectories from the state space of a RNN that predicts personality from financial transactions (Maree &

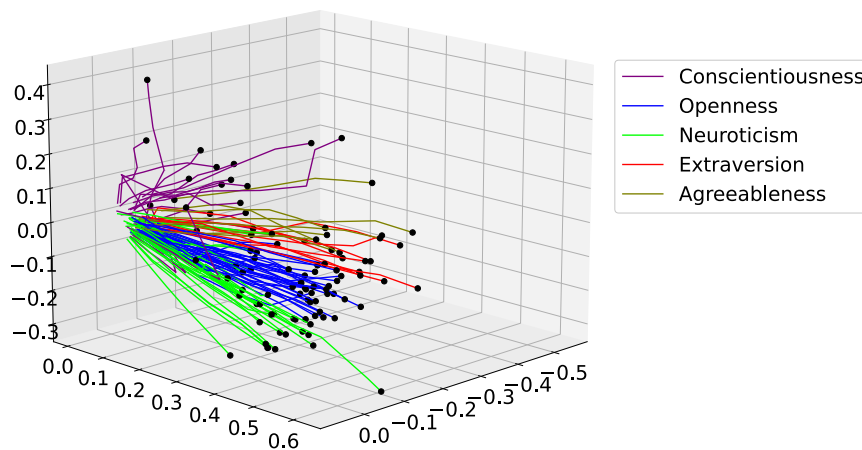
Omlin, 2021). We then detail our approach in combining these preliminaries to learn unique and personal compositions of prototypical strategies using hierarchical RL. Finally, we discuss our methodology of learning temporal strategies using several such compositions in a RNN. These temporal strategies eliminate the need to retrain orchestration agents when customers’ spending behavior change, or for new customers. We illustrate this process in a flow diagram in Fig. 1.

### 3.1 Personality-based profiling

We have previously developed a three-node RNN that predicts customer personalities from an input vector of their classified financial transactions (Maree & Omlin, 2021). This input vector consists of six annual time steps, each consisting of 97 transaction classes; the values in each time step add up to one and are the fraction of a customer’s annual spending per transaction category. The RNN output



**Fig. 1** Information flow diagram illustrating how our system uses financial transactions to generate personalized investment advice. We use hierarchical RL agents with intrinsic affinity to learn unique compositions of prototypical investment strategies that match personal financial preferences. We use many of these compositions to train a RNN to predict a final composition which allows for shifting strategies in time and eliminates the need to retrain an orchestration agent for each unique customer



**Fig. 2** Clustering behavior of a subset of 100 trajectories in the state space of a RNN. Each trajectory represents a customer’s spending behavior in time and is labelled according to the customer’s dominant personality trait. Each axis is the activation of one of the three nodes in the RNN

is a five-dimensional personality vector; its values are the degrees of membership in each of five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. We use the feature trajectories from this model’s state space—shown in Fig. 2—to represent a customer’s spending behavior over time.

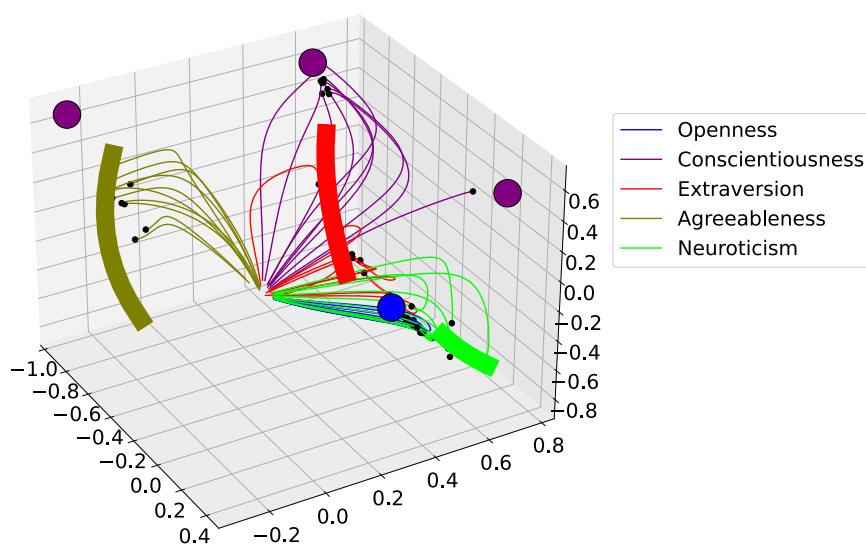
Each behavioral trajectory represents an individual customer and is labeled according to their most dominant personality trait: the trait with the greatest value in the personality vector. Linear trajectories indicate consistent spending behavior in time, while trajectories that veer from their initial direction indicate that that customer had changed their spending behavior. This explains why some trajectories seem to behave differently from others of the same color. We refer to Maree and Omlin (2022c) for a detailed discussion about the behavior of these trajectories. These trajectories form clusters in the state space, which separate into sub-clusters along the successive levels of lesser personality traits. This hierarchical clustering provides a means of micro-segmenting customers according to their spending behavior in time. We then explained these behavioral trajectories by reproducing them using a linear regression model, and we interpreted them through locating a number of attractors that govern the dynamics of the state space (Maree & Omlin, 2022c). We located these attractors by mapping the RNN output space into the state space through inverse regression. Using this mapping, and the maximum *reachable* values in the output space, based on the known range of the dimensions in the state space ( $[-1, 1]$ ), we extrapolated the final locations (attractors) of the behavioral trajectories. Formally:

$$\begin{aligned} \mathcal{O} &= \mathcal{D} \cdot \omega_{\text{inv}} - \vec{0} \cdot \omega_{\text{inv}} \\ \mathcal{D} &= \text{diag} \left\{ \max_{1 \leq i \leq |K|} \mathbf{O}_{i,j}, j \in [1..P] \right\} \\ \omega_{\text{inv}} &= (\mathbf{O}^T \mathbf{O})^{-1} \cdot (\mathbf{O}^T \mathbf{S}) \\ \mathbf{O} &= \mathbf{S} \cdot \omega_{\text{out}} \end{aligned}$$



where  $\mathcal{O} \in \mathbb{R}^{5 \times 3}$  is the projection of the output dimensions into the state space,  $\vec{0} \in [0]^P$  is the zero vector or origin of the output space,  $\mathcal{D} \in \mathbb{R}^{P \times P}$  is a diagonal matrix with the maximum values of each of the output dimensions on the diagonal,  $\mathbf{O} \in \mathbb{R}^{K \times P}$  is the matrix that holds the grid values of the reachable output space,  $\mathbf{S} \in [-1, 1]^{K \times 3}$  are the dimensions of the reachable state space,  $\omega_{\text{out}} \in \mathbb{R}^{3 \times P}$  is the matrix of weights of the RNN's output layer,  $P = 5$  is the number of output dimensions, and  $K$  is the number of points used to map the output hypercube. We corroborated these attractor locations with the observed destinations of the trajectories; we systematically chose different initial conditions in the state space and iterated the trajectories for 100 steps. We thus determined that trajectories converge towards the attractor associated with their most dominant personality trait. If a customer's spending behavior changes such that a different personality trait becomes dominant, their trajectory changes direction towards the new appropriate attractor. Figure 3 shows these attractors in the RNN state space, with the extended trajectories converging towards their corresponding attractors.

There are three point attractors for the personality trait conscientiousness, towards which trajectories converge depending on their initial conditions. Agreeableness, extraversion, and neuroticism each have a single line attractor, while trajectories that classified as openness converge towards a single point attractor. There is no distinction in significance between attractor types of the same class, in this case fixed attractors, nor is there a significance in the fact that one personality trait corresponds to three distinct point attractors (Ceni et al., 2019). Each basin of attraction forms a cluster of trajectories, which each form a hierarchy of sub-clusters along successive levels of dominance of personality traits. This is the interpretation of the trajectory dynamics.

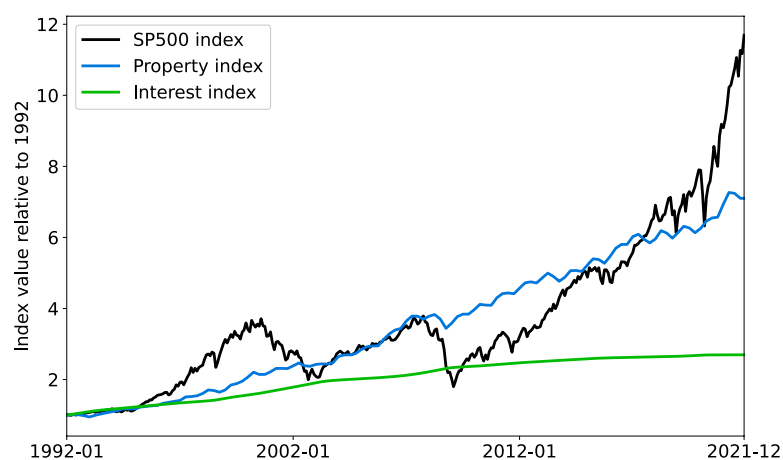


**Fig. 3** The locations of a set of attractors in the state space of a RNN. There are point and line attractors that are labelled according to the customers' corresponding dominant personality traits. We show 100 trajectories, with different initial conditions, asymptotically converging to their corresponding attractors

### 3.2 Learning prototypical investment strategies

Maree and Omlin (2022b) showed that interpretable RL can be used for investment that matches personality. In this preliminary study, we had trained multiple RL agents to invest monthly contributions in different financial asset classes: stocks, property, savings accounts, mortgage curtailment, and luxury items. While the investment classes stocks, property, and savings accounts are self-explanatory, we define mortgage curtailment as payments that reduce the principal balance of the loan, and luxury items as items defined in, e.g., the Knight-Frank luxury investment index (Knight Frank Company, 2022). There exists a trade-off in the allocation of funds between, e.g., mortgage curtailment and purchasing stocks: there are clear differences in expected risk and reward between these two strategies, which may appeal differently to different personality types. We obtained asset prices from the S &P 500 index (Yahoo Finance, 2022), the Norwegian property index (Statistics Norway, 2022), and the Norwegian interest rate index (Norges Bank, 2022) for the period between 1 January 1992 and 31 December 2021. We indexed these prices relative to their values on 1 January 1992 and plot these indices in Fig. 4.

With the help of a panel of banking experts from a major Norwegian bank, we ranked these asset classes according to a set of desirable asset class properties: high expected long-term returns, high perceived asset liquidity, low capital prerequisite, low expected long-term risk, and high perceived novelty. We based our assessment on known characteristics of each personality trait; (1) openness that values novelty and is drawn to change; (2) conscientiousness that is predisposed to planning and values structure; (3) extraversion that values having interesting topics for discussion; (4) agreeableness that values contributing to society; and (5) neuroticism that can more easily experience stress and anxiety (Tauni et al., 2017; Rizvi & Fatima, 2015). Our experts considered the relative affinities that each personality trait might have towards each of the asset class properties; they associated the personality traits with these properties, as shown in Table 1.



**Fig. 4** Asset pricing data for the S &P500 index, Norwegian property index, and Norwegian interest rate index. The values are relative to their respective values on 1 January 1992. These values are used in the state observations of our RL agents

**Table 1** Matrix *A* containing a set of asset class properties and their associations with the five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism

Asset class property	Open.	Cons.	Extra.	Agree.	Neur.
High returns	1	1	2	1	1
High liquidity	2	-1	2	1	2
Low capital prerequisite	0	-1	1	1	1
Low risk	-1	2	-1	1	2
High novelty	2	0	2	0	-1

The values are in the set  $\{n \in \mathbb{Z} \mid -2 \leq n \leq 2\}$  and indicate a strong negative, slightly negative, neutral, slightly positive and strong positive association, respectively

**Table 2** Matrix *B* containing ratings for the asset classes with regard to a set of properties

Asset class property	Savings	Property	Stocks	Luxury	Mortgage
High returns	0.25	0.67	1.00	0.05	0.50
High liquidity	1.00	0.25	0.80	0.10	0.05
Low capital prerequisite	0.80	0.25	1.00	0.50	1.00
Low risk	1.00	0.32	0.10	0.05	1.00
High novelty	0.10	0.25	0.75	1.00	0.10

The values are in the range  $[0, 1]$  and higher values represent higher performance in each of the asset class properties

**Table 3** Coefficients, in the range  $[-1, 1]$ , associating asset classes to prototypical personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism

Asset type	Open.	Cons.	Extra.	Agree.	Neuro.
Savings account	-0.11	0.08	-0.15	0.51	0.68
Property funds	-0.15	0.32	-0.22	-0.36	-0.24
Stock portfolio	0.82	-0.61	0.95	0.42	0.12
Luxury expenses	0.16	-0.51	-0.07	-0.80	-0.81
Mortgage repayments	-0.72	0.72	-0.52	0.23	0.25

Higher values indicate where personality traits might have higher affinities towards asset classes

The result showed that, for instance, the openness trait might highly value asset liquidity and novelty; because of their openness to new experiences, they might prefer to have cash readily at hand when such an opportunity presents itself, or they might value assets that in themselves may be perceived as novel.

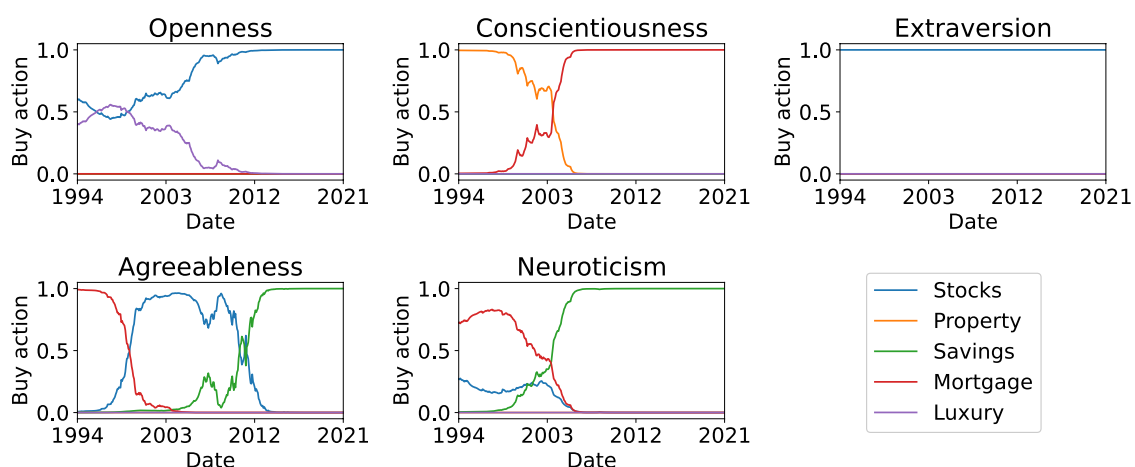
Another example is that the conscientiousness trait might prefer assets with low risk. The same panel of experts then ranked the asset classes according to the same set of properties, which we show in Table 2. We quantified risk and return from historical price data and the Sharpe ratio, respectively, and the values in Table 2 are normalized from these results.

We calculated a set of coefficients *C* that associate asset classes with personality traits using matrix multiplication:  $C = (A^T \cdot B^T)^T$ . These coefficients, scaled to the

range  $[-1, 1]$  and shown in Table 3, quantify personality-based affinities towards different asset classes.

These coefficients reveal that, for example, the extraversion trait has a high preference for stocks, whereas the conscientiousness agent prefers a combination of mortgage curtailment and property investment. This is in line with the findings of Gladstone et al. (2019) and Ramon et al. (2021). When scaled so that they add up to one and their minimum values are zero, these coefficients become the regularization priors  $\pi_0$  in Eq. (1); we regularized the objective functions of five prototypical agents to instill intrinsic affinities for certain asset classes. Each agent learned an investment strategy associated with one of the five personality traits, which is the interpretation of their policies. Figure 5 shows these strategies, where each agent acted in an environment in which it invested a fixed monthly amount of 10,000 Norwegian Kroner (NOK) over 28 years. The data included 30 years' pricing history between 1992 and 2022, but the first 24 months were used to initialize the RL environment variables: moving average convergence divergence (MACD) and relative strength index (RSI). Investments therefore started in 1994.

These prototypical agents clearly learned unique investment strategies. For simplicity, we did not include transaction costs, since investment happened with fixed amounts and frequencies (monthly); transaction costs are negligible and equal across comparisons. The openness agent initially preferred luxury items, in line with their openness to new experiences, and later purely invested in stocks, which had scored high in novelty. In contrast, the conscientiousness agent preferred to reduce risk through property investment, followed by a resolute mortgage curtailment. The prototypical agents' affinities and their long-term strategies are independent of market conditions and the duration of the investment period, because they are defined by constant priors (see Fig. 5). These are the low-level policies that we intend to



**Fig. 5** Action distributions of the five prototypical agents over a 30 year time period: between the ages of 30 and 60. Each figure represents the investment actions taken by one of the prototypical agents, who each associates with a single personality trait. Each line represents the fractional monthly investment into a labelled class of assets across the time period, e.g. the conscientiousness agent initially invests solely in property and subsequently in mortgage curtailment, while the extraversion agent consistently invests the entire monthly amount in stocks. A declining trend does not indicate selling of assets but rather a reducing monthly investment amount; the values on the y-axes are strictly positive indicating our agents never sell assets but rather change their monthly investment distributions

orchestrate into personalized investment strategies; customers have varying degrees of membership in each of the five personality traits, resulting in unique preferences for different asset classes that may change over time.

### 3.3 Hierarchical orchestration and temporal composition

We are inspired by the premise that there is a causal relationship between personality-matched spending and happiness (Matz et al., 2016). Tauni et al. (2017) provides empirical evidence that correlates personality to stock trading behavior, confirming earlier results from Rizvi and Fatima (2015). We, therefore, extend the concept of spending behavior in time to prosperity management. Our goal is to learn, through high-level RL orchestration, the optimum composition that match customers' unique financial personalities. Our RL agent orchestrates the actions of low-level prototypical agents according to customers' extracted behavioral trajectories (Fig. 2). With actions adding up to one, representing the fraction of the investment amount allocated to each low-level agent, it maximizes the following reward function:

$$R = \vec{H} \cdot (\vec{P} \cdot \mathbf{C}), \quad (2)$$

which is the dot product between the current values of asset class holdings  $\vec{H}$  and the customer's preference for each asset class. This preference is the dot product of the customer's personality vector  $\vec{P}$ , i.e., the set of five values representing their degrees of membership in each of the personality traits, and the set of coefficients  $\mathbf{C}$  that relate each asset class with each personality trait (Table 3). The dot product is a scalar value that represents a customer's association with each asset class. By adding the associations of each personality trait with the different asset classes, multiplied by a customer's fuzzy degree of membership in the personality trait, we estimate the customer's association with each asset class. This reward measures the correlation between spending behavior and investment strategy, which we call the *satisfaction index*; the higher the satisfaction index, the higher the correlation between spending behavior and investment strategy. A limitation of this metric is that it is not a fair performance comparison of different customers with different personality profiles; the satisfaction indices will be different between one customer with a perfectly conscientious profile and portfolio and one with a perfectly extraverted profile and portfolio. It is, however, a metric that enables comparison between different methods of composing a strategy for a given customer, and that is how we use it. We then use the regularization prior:

$$\pi_0 = \frac{\vec{P}}{\sum \vec{P}}, \quad (3)$$

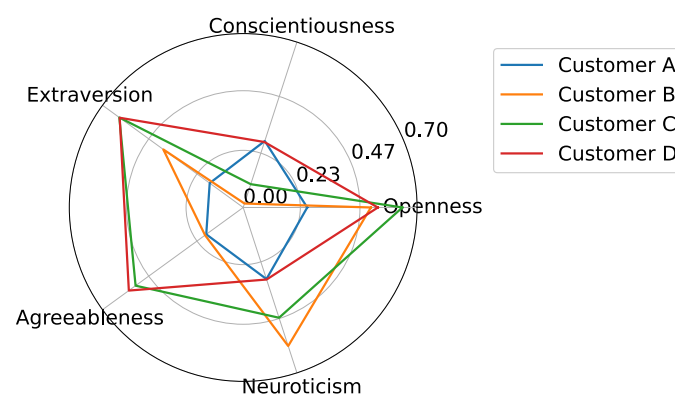
to instill an intrinsic RL affinity in a set of DDPG agents. The actor consisted of three vanilla RNN nodes and an output layer of five actions with a softmax activation. The critic had a similar three-node RNN layer for the states which, concatenated with the actions, were succeeded by a 1000 node feed-forward layer and a single output

node with no activation function. We found that three RNN nodes consistently provided high total rewards, which is consistent with findings that RNN architectures generally perform well in low-dimensional representations (Maheswaranathan et al., 2019). We tuned our hyperparameters using a one-at-a-time parameter sweep to reach the following optima: the actor and critic learning rates were 0.005 and 0.01, respectively, the target network update parameter  $\tau$  was 0.05, the discount factor  $\gamma$  was 0.95, and the regularization scaling factor  $\lambda$  was 5.

Finally, we trained a RNN to predict this composition of prototypical agents from a sample of 500 pre-trained orchestration agents; the orchestration agents learned their strategies, as described above, from the data for 500 unique customers of a major Norwegian bank. We used the customers' feature trajectories—their encoded spending behavior—as input to a neural network with three RNN nodes. The output from this network is the actions of the orchestration agents, i.e. the unique, locally optimal combination of the five prototypical agents. We used 400 customers for training and 100 for testing. The learning rate was 0.0005 and the model typically converged within 10,000 iterations. We used this model to predict the composition of prototypical agents for customers as their spending behavior varies in time; the RNN uses a rolling window of 6 years' spending behavior.

## 4 Results

In this section, we compare hierarchical RL to simple linear combinations of the prototypical agents; our agents find locally optimum compositions with similar financial returns, but improved personalization compared to simple linear combinations. We also demonstrate how these compositions can accommodate changing spending behavior in time; financial personalities may fluctuate in time, and our system adapts in a non-erratic way.



**Fig. 6** The personality vectors representing the personality traits of four real customers. Each colored line represents a customer and each axis on the radar plot represents a personality trait. The values on the axes are in the range [0, 1] and represent the customers' degree of membership in each of the personality traits. These customers were selected to represent a range of personality profiles: Customer A has a balanced profile, Customer B scores high in neuroticism and openness, Customer C scores high in openness and extraversion, and Customer D scores high in extraversion, agreeableness, and openness

## 4.1 Hierarchical orchestration of prototypical agents

For illustration purposes, we selected four customers from a major Norwegian bank for whom we trained personal orchestration agents; their personality vectors, visualized in Fig. 6, were derived from their historical financial transactions using the RNN described in Sect. 3.1. They were chosen to represent a range of personality profiles.

Customer A has a relatively balanced profile, with low variation in the values of their personality vector, which also has relatively small values. This contrasts with Customer B who scores high in neuroticism and openness, Customer C who scores high in openness and extraversion, and Customer D who scores high in extraversion, agreeableness, and openness. Their respective regularization priors are shown in Table 4.

The regularization prior for Agent A  $\pi_{0,A}$  (the agent for Customer A) consequently has a low variation in its values while  $\pi_{0,B}$  assigned the highest weight to neuroticism and openness,  $\pi_{0,C}$  assigned the highest weight to openness and extraversion, and  $\pi_{0,D}$  assigned the highest weight to extraversion, agreeableness, and openness.

These four customers' personality profiles, and consequently the orchestration agents' actions, were constant in time. Customers' personality profiles may naturally vary in time, causing directional changes in their behavioral trajectories, which alter the orchestration agent's action distribution. We will discuss the effects of time-variant customer spending behavior on the compositions in Sect. 4.2. The investment strategies for the four customers are shown in Fig. 7.

Although these strategies might seem similar, there are significant differences: Customer A never invested more than 60% of their monthly allocation in stocks, while Customer D invested up to 90% in stocks, and Customer A was the only one to invest significantly in property. This is due to Customer A having the highest relative degree of conscientiousness, i.e., they preferred a reduced risk. In contrast, Customer D had the highest risk in their portfolio by investing the least in property and mortgage curtailment and the most in stocks, due to their low score in neuroticism which increases their appetite for risk. When comparing Customers B and C, Customer B invested more in savings accounts and less in stocks in the period between 150 and 250 months. This is due to their differences in agreeableness and neuroticism, where customer B scored higher in neuroticism and lower in agreeableness. In Fig. 5, the prototypical agents associated with neuroticism and agreeableness are

**Table 4** Regularization priors used during training of the orchestration agents of four customers, named A through D

Prior	Open.	Cons.	Extra.	Agree.	Neur.
$\pi_{0,A}$	0.22	0.24	0.14	0.15	0.25
$\pi_{0,B}$	0.30	0.01	0.23	0.11	0.35
$\pi_{0,C}$	0.27	0.04	0.26	0.23	0.20
$\pi_{0,D}$	0.23	0.12	0.27	0.25	0.13

Each row represents the regularization prior  $\pi_{0,i}$  for one of the orchestration agents  $i \in [A, D]$ . The values are in the range  $[0, 1]$  and add to one for each prior. They represent the fraction of investment amount allocated to each prototypical low-level agent: openness, conscientiousness, extraversion, agreeableness and neuroticism. A higher values indicates a higher weighting of that agent's strategy



**Fig. 7** Investment advice from four personal investment agents for four different customer personalities; they are the combined actions of the prototypical agents according to the orchestration agent. Each plot shows the investment advice in time for a single customer, named “Customer A” through “Customer D” in accordance with the labels in Fig. 6. Declining trends do not indicate selling of assets but rather reduced monthly investment in that asset; the values on the y-axes are strictly positive indicating that assets are never sold, but investment distributions change across assets

the only two to invest in savings, and the neuroticism agent started investing in savings much earlier and with higher percentages. Despite the nuanced differences in investment approaches, the general advice for all customers was similar: first pay down mortgages to reduce debt repayments, then accept higher risk with higher returns from stocks and benefit from compound growth, and finally toward retirement age reduce risk through savings accounts. This is consistent with conventional financial advice: younger people with more disposable income may accept more risk for higher returns. Very interestingly, this was not explicit in the objective function, which had no elements of risk, while the effect of compound growth was evident only in increased final returns.

The monthly investments accumulated to 3.36 million NOK, and the portfolios were initiated with a 2 million NOK property investment with a corresponding 2 million NOK mortgage; individual strategies may vary between, e.g., quickly reducing the principal balance of the loan thus avoiding interest or investing in more risky asset classes such as stocks. The theoretical maximum return was 27.7 million NOK, achieved when investing purely in stocks. The final financial returns for our four customers were very similar: after 28 years of investing 10,000 NOK per month, they all had portfolio values ranging between 21 and 24 million NOK. However, our aim was to optimize customer satisfaction in their portfolio while still achieving high returns. We note that the satisfaction index is not a suitable metric for comparing different customers, and this is reflected in the results, where satisfaction indices between customers had greater variation than their financial returns. However, we compare the satisfaction index between different compositions of prototypical agents for the same customer: Table 5 shows the results of the orchestration agents and those of a linear combination of the prototypical agents.



**Table 5** Performance metrics comparing the orchestration agent to a simple linear combination of the prototypical agents

Customer	Orchestration agent			Linear combination		
	Portfolio value mill. NOK	Satisfac- tion index	Sharpe ratio	Portfolio value mill. NOK	Satisfac- tion index	Sharpe ratio
A	20.9	3.1	0.4393	20.9	3.0	0.4367
B	22.0	12.9	0.3704	21.7	12.7	0.3730
C	22.7	19.0	0.3528	22.4	17.8	0.3616
D	23.8	19.5	0.3350	22.6	14.5	0.3706

We list the resulting portfolio values and satisfaction scores for both these scenarios after investing 10,000 NOK per month for 28 years according to the strategies shown in Fig. 7. Here, the Sharpe ratio is the mean of the monthly returns divided by the standard deviation of the monthly returns

This linear combination is the dot product of the personality vector and the action vectors of the prototypical agents, scaled such that the resulting actions add up to one; the actions of the prototypical agents were weighted according to customers' personality vectors. In terms of profit and satisfaction index, the orchestration agent never performs worse than a linear combination of prototypical agents; although it typically achieves only slightly better financial returns, it can significantly improve the satisfaction index. This was not the case when using feed-forward networks to process the customer spending input, which returned inconsistent results across multiple training runs and frequently performed worse than the simple linear combination. This is consistent with findings from (Tovanich et al., 2021) that spending patterns in time hold salient information not evident in non-temporal data. The Sharpe ratios are similar between the customers' orchestration agents and linear combinations, and only Customer A had a higher Sharpe ratio for the orchestration agent. This is explained by Customer A's relatively high score in conscientiousness which, as stated before, resulted in increased investment in property—a lower risk asset class—and a corresponding reduction in portfolio risk. We calculated the Sharpe ratio for the global optimum strategy—investing solely in stocks—as 0.2856 indicating an increased risk in the portfolio. This strengthens our argument that our locally optimal personalized strategies could be improvements over the global optimum in returns.

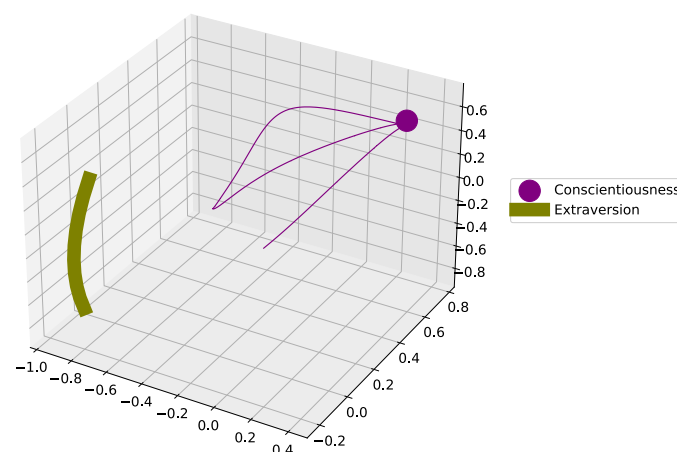
We regularized the orchestration agents to act according to a specified prior with the same action distribution as for the linear combination scenario. Through stochastic gradient descent, they optimized the satisfaction index in that region of the action space. In Fig. 11, we illustrate the policy convergence towards local optima of each of the four orchestration agents. The policies were randomly initialized, but quickly converged to local optima in close proximity to the regularization priors in the action space. The learned strategies are thus interpretable.

## 4.2 Time-variant analysis

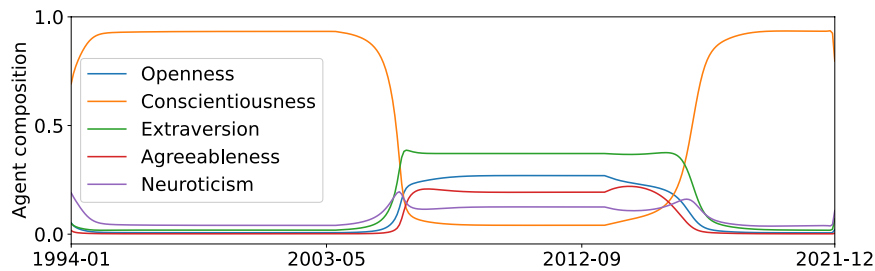
We have access to historical transactions dating back a maximum of 6 years, which hinders long-term time-varying analyses of customers' spending behavior. However,

we created a fictitious customer, Customer E, by copying the financial transactions of two distinct customers: one who scored high in conscientiousness and another who had a slightly more balanced profile while demonstrating mostly extraverted spending behavior. We constructed Customer E's transaction history as follows: we duplicated 1 year's transactions from the conscientious customer 10 times, then 1 year's transactions from the extraverted customer 10 times, and the final 8 years' transactions again from the conscientious customer. Customer E thus exhibited 10 years of conscientious spending behavior, followed by 10 years of mixed, but mostly extraverted behavior, followed by the final 8 years of conscientious behavior once again. Our aim was to demonstrate what effect a change in spending behavior has on the investment strategy. Figure 8 shows the encoded spending behavior, or the feature trajectory, of this fictitious customer. It follows the expected behavior and converges towards the corresponding conscientiousness and extraversion attractors. It is not expected that a trajectory converges exactly on top of an attractor with every change in spending behavior but that it moves towards the corresponding attractor. This illustrates the interpretation of our feature extraction model: by observation and with knowledge of the locations of the attractors that govern the dynamics of the system, we can reason about the functioning of the model.

We trained a RNN from the spending behavior of 500 customers from a major Norwegian bank to predict the actions of their corresponding orchestration agents. Using this RNN, we predicted the recommended composition of prototypical agents for Customer E, shown in Fig. 9. This investment strategy highly favors the conscientiousness agent in the first 10 years, after which the composition changes to a mixture of agents that is biased towards extraversion. This transition does not happen immediately and there is a gradual shift over the course of a few years. This is important, as financial advice should not be erratic. The mixture of agents is



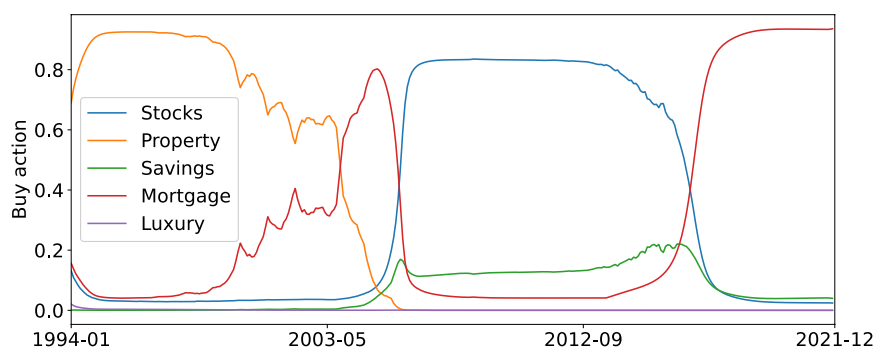
**Fig. 8** The encoded spending behavior for a fictitious customer, Customer E, drawn as feature trajectories in the state space of the RNN from Fig. 2. This customer's financial transactions were such that their spending personalities were first predominantly conscientious, then extraverted, and finally conscientious once again. We show the two corresponding attractors and the customer's trajectory which initially converges on the conscientiousness attractor. As soon as the customer's spending pattern changes, the trajectory moves towards the corresponding new attractor: extraversion. Finally, and before a sufficient time has passed for the trajectory to converge on the new attractor, the spending pattern changes back to conscientiousness and the trajectory once again converges on that attractor



**Fig. 9** The recommended composition of prototypical agents for Customer E. We created Customer E to display highly conscientious spending behavior between 1994 and 2004. Between 2005 and 2015, they displayed spending behavior related to a mixed personality profile which was mostly extraverted. From 2015 onward, their spending was once again conscientious. This time-varying spending behavior is reflected in the weights assigned in the composition of prototypical agents: conscientious spending behavior results in a conscientious investment strategy, which can change in time with changing spending behavior

expected and can be explained by observing the spending trajectory in Fig. 8: the trajectory has not yet converged to the extraversion attractor and may fall close to the basin of attraction of several other personality attractors. It also corresponds to the behavior of the selected customer from whom we copied transactions: they were predominantly extraverted but also showed behavior from other traits such as openness, agreeableness, etc. This result shows that while the dominant personality trait is important—extraversion is the largest portion of the composition—our system also considers other traits. In the last 8 years, the composition shifts back to favoring the conscientiousness agent.

Figure 10 shows the composed strategy for Customer E which, unsurprisingly, closely follows the prototypical conscientiousness strategy in the initial and final phases, while in the middle it invests more in stocks and savings accounts. We show the portfolio value and asset class holdings in Figs. 12 and 13 respectively. While



**Fig. 10** The long-term, time-variant investment strategy for a fictitious customer, Customer E. We created Customer E to display highly conscientious spending behavior between 1994 and 2004. Between 2005 and 2015 they displayed spending behavior related to a mixed personality profile which was mostly extraverted. From 2015 onward their spending was once again conscientious. The investment strategy, according to the time-variant composition of the prototypical agents (Fig. 9), is related to the customer's spending behavior in time. Initially, the conscientious spender invests conscientiously—in low risk asset classes, namely property—while between 2005 and 2015 the extraverted spender invests mostly in stocks with an element of agreeableness evident in their investment in savings accounts. Finally, the strategy reverts to a conscientious behavior and resolute mortgage curtailment

investment in stocks corresponds to strategies from extraversion, openness, and agreeableness, investment in savings are related to the agreeableness strategy. This strategy is clearly interpretable from the perspective of spending behavior in time. From a customer's financial records, we can estimate their spending personality and extract behavioral features using an RNN. We can reason about these features based on their locations and trajectories in the state space of our RNN.

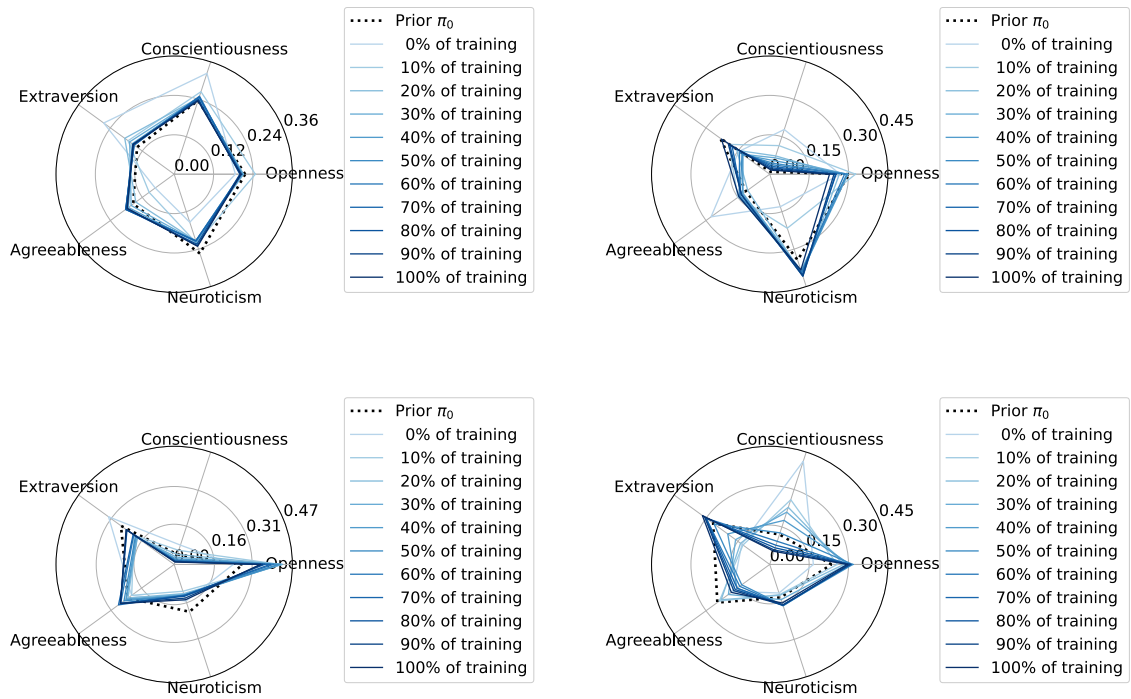
Then, we can combine the actions of five, interpretable, prototypical agents to suggest an investment strategy. We can also reason about this strategy given the inherent affinities of our prototypical agents. This ability to reason about the predictions of a system inspires trust and removes a cloak of uncertainty.

## 5 Conclusions

Machine learning is essential for personalizing financial services. Its acceptance is contingent on understanding the underlying models, which makes model explainability and interpretability imperative. Our reinforcement learning model blends investment advice that is aligned with different personality traits. Its interpretation follows from the global intrinsic affinities of the learned policies, i.e., affinities that are independent of the current state. These policies not only result in good profit, but also similar profits are achieved across different personality profiles despite their distinct strategies. For instance, they avoid risk for highly conscientious individuals, while pursuing novelty for individuals that are more open to new experiences. Their time-variant strategies adapt in a non-sporadic way to changes in spending behavior. Interestingly, our agents have learned the concept of risk without this being explicit in the objective function. Across all portfolios, the advice is consistent with conventional wisdom: younger investors may accept higher risk, which typically reduces with age. It remains to be seen whether this is simply a consequence of optimizing profit while balancing the intrinsic action distribution, or whether our agents have learned deeper strategies of asset management. In future work, we intend to investigate this phenomenon by extracting an explanation for our agents' decisions. It will also be interesting to extend our method to local intrinsic affinity, where the preferred policy also depends on the *current state*. It is compelling to generalize the approach by Nangue Tasse et al. (2020) who decompose tasks and suggest a Boolean algebra for the composition of the learned strategies; ours is a fuzzy composition of prototypical agents that might benefit from such an extension. The potential applications for our method go beyond investment advice and include, e.g., autonomous vehicles, personalized teaching and learning, treatment of chronic diseases, or the design of virtuous agents in the context of artificial morality.

## Appendix 1: Training convergence

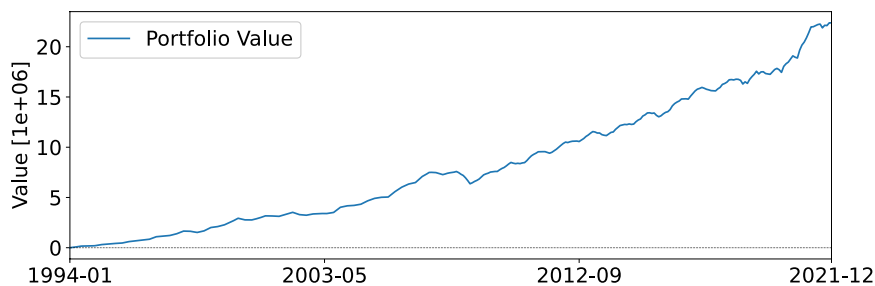
See Fig. 11.



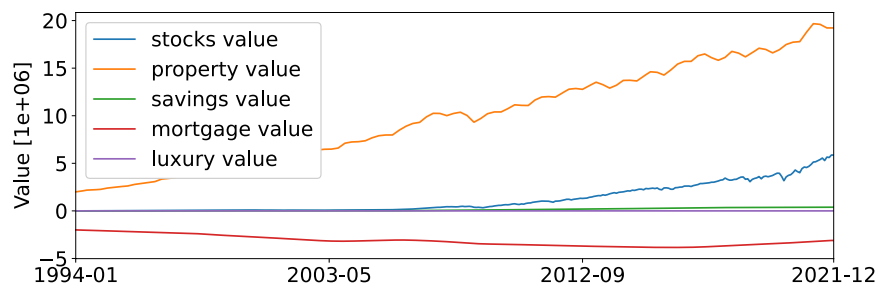
**Fig. 11** Training convergence of the orchestration agents for four different customer personality profiles. Each successively darker blue line represents the orchestration action distribution after an increasing number of training runs. As training progresses, the successively darker blue lines converge towards the learned action distribution. The black dotted line represents the regularization prior  $\pi_{0,i}$ . The figures show how randomly initialized policies converge towards their specified priors and settle in a local optima in close proximity

## Appendix 2: Example portfolio returns

See Figs. 12 and 13.



**Fig. 12** The resulting portfolio value for Customer E, a fictitious customer designed to illustrate the time-varying investment strategy for a customer whose spending behavior varies in time. Customer E first exhibited conscientious spending behaviour, followed a period of extraverted behavior with significant elements from other traits, and finally they reverted to conscientious spending. The portfolio value follows an upward trend with a slight downward variability in about 2008. The reason for this contraction becomes evident when combining information from Figs. 13 and 4: the customer has a relatively high holding in property for which there was a market contraction in 2008



**Fig. 13** The resulting portfolio value for Customer E, a fictitious customer designed to illustrate the time-varying investment strategy for a customer whose spending behavior varies in time. Customer E first exhibited conscientious spending behaviour, followed a period of extraverted behavior with significant elements from other traits, and finally they reverted to conscientious spending. The asset class holdings correspondingly favours property initially and this asset class experiences compound growth throughout the investment period following its index shown in Fig. 4. The strategy only invests in stocks between about 2005 and 2017 (refer to Fig. 10) and the stock holding is correspondingly low

**Funding** Open access funding provided by University of Agder. This study was partially funded by a grant from The Norwegian Research Council, project number 311465.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Ethics approval** Not applicable.

**Consent to participate** Personal data were anonymized and processing was done on the basis of consent in compliance with the European General Data Protection Regulation (GDPR).

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Andres, A., Villar-Rodriguez, E., & Ser J. D. (2022). Collaborative training of heterogeneous reinforcement learning agents in environments with sparse rewards: What and when to share? [arXiv:2202.12174](https://arxiv.org/abs/2202.12174)
- Apeh, E. T., Gabrys, B., & Schierz, A. (2011). Customer profile classification using transactional data. *2011 Third World Congress on Nature and Biologically Inspired Computing* (pp. 37–43). Salamanca, Spain.
- Aubret, A., Matignon, L., & Hassas, S. (2019). A survey on intrinsic motivation in reinforcement learning. [arXiv:1908.06976](https://arxiv.org/abs/1908.06976)
- Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, *104*(3), 671–732.

- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., et al. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.
- Beyret, B., Shafti, A., & Faisal, A. (2019). Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5014–5019). Macau, China.
- Cao, L. (2021). AI in finance: Challenges, techniques and opportunities. *Banking & Insurance eJournal*, 55, 1–38.
- Ceni, A., Ashwin, P., & Livi, L. F. (2019). Interpreting recurrent neural networks behaviour via excitable network attractors. *Cognitive Computation*, 12, 330–356.
- Fernández, A. (2019). Artificial intelligence in financial services. Tech. rep., The Bank of Spain, Madrid, Spain.
- Galashov, A., Jayakumar, S., Hasenclever, L., et al. (2019). Information asymmetry in KL-regularized RL. *International Conference on Learning Representations (ICLR)* (pp. 1–25). New Orleans: Louisiana, United States.
- García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42), 1437–1480.
- Gladstone, J. J., Matz, S. C., & Lemaire, A. (2019). Can psychological traits be inferred from spending? Evidence from transaction data. *Psychological Science*, 30(7), 1087–1096.
- Hengst, B. (2010). *Hierarchical reinforcement learning* (pp. 495–502). Springer.
- Heuillet, A., Couthouis, F., & Díaz-Rodríguez, N. (2021). Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214(106685), 1–24.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Knight Frank Company (2022) Knight Frank luxury investment index. <https://www.knightfrank.com/wealthreport/luxury-investment-trends-predictions/>. Accessed 27 May 2022.
- Kulkarni, T. D., Narasimhan, K., Saeedi, A., et al. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 29, pp. 1–9). Curran Associates Inc.
- Levy, A., Platt, R., & Saenko, K. (2019). Hierarchical reinforcement learning with hindsight. In: *International conference on learning representations* (pp. 1–16).
- Lillicrap, TP., Hunt, JJ., & Pritzel A., et al. (2019). Continuous control with deep reinforcement learning. [arXiv:1509.02971](https://arxiv.org/abs/1509.02971).
- Maheswaranathan, N., Williams, A. H., Golub, M. D., et al. (2019). Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics. *Advances in Neural Information Processing Systems (NIPS)*, 32, 15696–15705.
- Maree, C., & Omlin, C. W. (2021). Clustering in recurrent neural networks for micro-segmentation using spending personality. In: *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–5).
- Maree, C., & Omlin, C. W. (2022). Understanding spending behavior: Recurrent neural network explanation and interpretation (in print). In: *IEEE computational intelligence for financial engineering and economics* (pp. 1–7).
- Maree, C., & Omlin, C. (2022). Reinforcement learning your way: Agent characterization through policy regularization. *AI*, 3(2), 250–259.
- Maree, C., & Omlin, C. W. (2022). Can interpretable reinforcement learning manage prosperity your way? *AI*, 3(2), 526–537.
- Marzari, L., Pore, A., Dall’Alba, D., et al. (2021). Towards hierarchical task decomposition using deep reinforcement learning for pick and place subtasks. *20th international conference on advanced robotics (ICAR)* (pp. 640–645). Ljubljana, Slovenia.
- Matz, S. C., Gladstone, J. J., & Stillwell, D. (2016). Money buys happiness when spending fits our personality. *Psychological Science*, 27(5), 715–725.
- Millea, A. (2021). Deep reinforcement learning for trading: A critical survey. *Data*, 6(11), 1–25.
- Milnor, J. (2004). *On the concept of attractor* (pp. 243–264). Springer.
- Miryoosefi, S., Brantley, K., & Daume, III H., et al. (2019). Reinforcement learning with convex constraints. In: *Advances in neural information processing systems* (pp. 1–10).
- Mousaeirad, S. (2020). Intelligent vector-based customer segmentation in the banking industry. [arXiv:2012.11876](https://arxiv.org/abs/2012.11876).

- Nangue Tasse, G., James, S., & Rosman, B. (2020). A Boolean task algebra for reinforcement learning. *34th conference on neural information processing systems (NeurIPS 2020)* (pp. 1–11). Vancouver, Canada.
- Norges Bank. (2022). Interest rates. <https://app.norges-bank.no/query/#/en/interest>. Accessed 30 Jan 2022.
- Pateria, S., Subagdja, B., Tan, Ah., et al. (2021). Hierarchical reinforcement learning: A comprehensive survey. *Association for Computing Machinery*, 54(5), 1–35.
- Ramon, Y., Farrokhnia, R., Matz, S. C., et al. (2021). Explainable AI for psychological profiling from behavioral data: An application to big five personality predictions from financial transaction records. *Information*, 12(12), 1–28.
- Rizvi, S., & Fatima, A. (2015). Behavioral finance: A study of correlation between personality traits with the investment patterns in the stock market. *Managing in Recovering Markets* (pp. 143–155). New Delhi: Springer India.
- Smith, W. R. (1956). Product differentiation and market segmentation as alternative marketing strategies. *Journal of Marketing*, 21(1), 3–8.
- Statistics Norway. (2022). Table 07221-Price index for existing dwellings. <https://www.ssb.no/en/statbank/table/07221/>. Accessed 30 Jan 2022.
- Stefanel, M., & Goyal, U. (2019). *Artificial intelligence & financial services: Cutting through the noise*. APIS partners, London, England: Tech. rep.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). The MIT Press.
- Tauni, M. Z., Rao, Zu. R., Fang, H., et al. (2017). Do investor's big five personality traits influence the association between information acquisition and stock trading behavior? *China Finance Review International*, 7(4), 450–477.
- Tovanich, N., Centellegher, S., Bennacer Seghouani, N., et al. (2021). Inferring psychological traits from spending categories and dynamic consumption patterns. *EPJ Data Science*, 10(24), 1–23.
- Vieillard, N., Kozuno, T., & Scherrer, B., et al. (2020). Leverage the average: An analysis of KL regularization in reinforcement learning. In: *Advances in Neural Information Processing Systems (NIPS)* (vol. 33, pp. 12163–12174). Curran Associates.
- Yahoo Finance. (2022). Historical data for S &P500 stock index. <https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC>. Accessed 30 Jan 2022.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.





# Appendix H

## Symbolic Explanation of Affinity-Based Reinforcement Learning Agents with Markov Models

This paper has been submitted as:

C. Maree and C. W. Omlin, “Symbolic Explanation of Affinity-Based Reinforcement Learning Agents with Markov Models”, Submitted to *Expert Systems with Applications*, **2022**, doi: 10.48550/arXiv.2208.12627.

Copyright © 2022 Elsevier

# Symbolic Explanation of Affinity-Based Reinforcement Learning Agents with Markov Models

Charl Maree<sup>a,b,\*</sup>, Christian W. Omlin<sup>a</sup>

<sup>a</sup>*Center for Artificial Intelligence Research, University of Agder, Grimstad, 4879, Norway*

<sup>b</sup>*Chief Technology Office, Sparebank 1 SR-Bank, Stavanger, 4007, Norway*

---

## Abstract

The proliferation of artificial intelligence is increasingly dependent on model understanding. Understanding demands both an interpretation—a human reasoning about a model’s behavior—and an explanation—a symbolic representation of the functioning of the model. Notwithstanding the imperative of transparency for safety, trust, and acceptance, the opacity of state-of-the-art reinforcement learning algorithms conceals the rudiments of their learned strategies. We have developed a policy regularization method that asserts the global intrinsic affinities of learned strategies. These affinities provide a means of reasoning about a policy’s behavior, thus making it inherently interpretable. We have demonstrated our method in personalized prosperity management where individuals’ spending behavior in time dictate their investment strategies, i.e. distinct spending personalities may have dissimilar associations with different investment classes. We now explain our model by reproducing the underlying prototypical policies with discretized Markov models. These global surrogates are symbolic representations of the prototypical policies.

*Keywords:* Explainable AI, personalized financial services, policy regularization, affinity-based learning, Markov models

---

\*Corresponding author. Email address: charl.maree@uia.no

## 1. Introduction

The ultimate goal of explainable AI is understanding. It builds trust, improves safety, and improves predictive performance by facilitating precise model improvements [1]. For instance, feature saliency analyses can improve feature selection and consequently the predictive performance in stock trading [2], and rule extraction can enhance trust in an AI system for loan approvals Sachan et al. [3]. Despite considerable advancement in fields such as explainable reinforcement learning (RL) [4, 5, 6], the explainability of their underlying models has not yet been fully addressed [7, 8].

Reinforcement learning has become omnipotent in finance, for example, multi-agent RL for algorithmic trading [9]. Methods such as probabilistic argumentation [10], structural causal modeling [11], and introspection through interesting elements [12] exemplify the pursuit of post-hoc explainability. We, however, propose an alternative approach: rather than attempting to extract the learned strategy post hoc, ours is an intrinsic method that instills a desirable behavior during training [13]. Through regularization of the objective function, our method encourages global action affinities and thus exercises control over what agents learn. We have demonstrated the value of our method in personal prosperity management, where individual spending behaviors dictate investment strategies [14]. We instilled affinities for certain asset classes into the policies of a set of prototypical agents, each associating with a given personality trait. For example, a conscientiousness agent prefers asset classes typically associated with reduced risk.

Understanding ensues from a model explanation and an interpretation of its behavior. We distinguish between these two concepts: an explanation is a symbolic representation of a model’s predictions, while an interpretation is a human reasoning about its behavior. While our agent’s policies are inherently interpretable, they lacked a symbolic explanation. Using discretized Markov models, we now provide that explanation and thus gain insight into previously unanswered questions, such as why do the agents invest according to conventional wisdom: exploiting the benefits of compound growth and reducing risk with increasing customer age. These previously unanswered questions demonstrate the need for both explanations and interpretations: the lack of a symbolic representation of agents’ policies inhibited our complete understanding.

Our contributions are therefore: (1) we demonstrate how to instill global action affinities, thus affecting how RL agents learn, which we argue is a useful

paradigm shift over the current approach of either post hoc rule extraction or constrained learning, (2) we distinguish between model explainability and interpretability, and in an empirical example demonstrate the difference and the utility of both, and (3) we propose a method of using Markov models to extract symbolic explanations of RL agents’ policies. In the next section, we provide an overview of the current state of the art in explainable RL and identify limitations in the field. We then describe our data and empirical methodology, discuss our results, and conclude with insights and future work.

## 2. Related Work

RL agents learn to solve problems by maximizing the total expected reward awarded by the environment in which they act. They are particularly adept at learning in the presence of sparse and delayed rewards [15]. The environment is a discrete-time process where the current state depends only on the previous state and the action taken by the agent: a Markov decision process (MDP), described by the tuple  $(S, A, R, P)$  where  $S$  is a set of states,  $A$  a set of actions,  $R(s, a)$  the reward for taking action  $a \in A$  in the state  $s \in S$ , and  $P(s, a) = P(s'|s, a)$  the probability that action  $a$  in the state  $s$  leads to the state  $s'$  [16]. Deep deterministic policy gradients (DDPG) is a model-free RL algorithm for learning policies in a continuous action space [17]. A DDPG agent consists of four neural networks: an actor  $\mu(\theta)$  representing the policy, a critic  $Q(\theta)$  representing the state action value function, and for numerical stability, a target actor  $\mu'(\theta')$  and a target critic  $Q'(\theta')$ . During learning, the target network parameters are typically updated slowly given a soft update parameter  $\tau \in [0, 1]$  with a small value:  $\theta'_i = \tau\theta_i + (1 - \tau)\theta'_i$ ,  $i \in \{\mu, Q\}$ .

Explainable RL has traditionally employed generic methods that explain the underlying models of agents [1]. More recently, however, bespoke methods have emerged that consider the state-action space and / or the behavior of the learned policy [6, 4, 5]. Most, if not all, of these approaches extract explanations after training; they generalize the learned policy through observation or statistical analyses. Few of these extracted explanations match our definition of explainability, and most are more accurately described as interpretations. *State representation learning* connects the state space with information from actions, rewards, or expert knowledge when extracting representations that are useful for reasoning about policies [18]. Under certain restrictions, e.g., linearity, it learns models that either predict states from state-action pairs, or actions from states, thus simplifying the state-action

space and improving interpretability. *Introspection* analyzes an agent’s experience through statistics such as the frequency of occurrences of states, state-actions, and transitions, the transition probabilities, and estimated rewards compared to the learned state-action value function [12]. It uses interesting elements from this analysis, such as outliers, mean values, etc. to reason about agents’ behaviors. *Structural causal modeling* learns causal relationships between states, actions, and rewards by defining action influence graphs that map the action transitions for all possible paths from an initial state to a set of terminal states [11]. It defines the causal chain as the one path in the action influence graph that matches the learned policy, and a reward chain as the vector of rewards along this causal chain. Its interpretation of the policy is the comparison between the reward chain and all other possible reward vectors that do not follow the causal chain. *Probabilistic argumentation* uses argumentation graphs—sets of attacking and supporting arguments for each action in a finite action space—to learn interpretations in a RL setting [10]. The state is the intersection of the argumentation graph and the policy to be explained, the actions form a probabilistic distribution across the arguments, and the rewards depend on whether an argument attacks or supports the current action. The learned policy provides probabilistic interpretations of agents’ actions in human understandable terms: supporting and attacking arguments for each action. *Reward decomposition* replaces the scalar reward with a vector of more meaningful rewards, where the total reward is the sum of the vector [19, 20, 21]. Although evaluating the reward vector for a given action might enable reasoning about that action in meaningful terms, it does not take into account expected future rewards and can be insufficient in environments with delayed or sparse rewards. *Reward redistribution* addresses this problem by redistributing delayed rewards in time; it assigns credit to previous actions, thus reducing the delay of the reward [22]. The immediate reward for each time step in a sequence of state-action transitions is equal to the change in the total expected reward. It defines key interpretable events in the policy and, through sequence alignment, redistributes rewards to those events given a set of transition sequences. *Hierarchical RL* divides complex tasks into smaller and simpler tasks that are solved by correspondingly simpler RL agents [23, 21]. An orchestration agent learns to sequentially combine these prototypical agents to solve complex tasks. If tasks are sufficiently subdivided, the interpretation, or human reasoning about agents’ decisions, follows from their simplicity.

The complexity of RL models exacerbates the issue of fidelity and vali-

dation of any post hoc explanation. We, instead, encourage agents to adapt their behavior during learning, thus instilling an inherent probabilistic action affinity that is also an interpretation of their behavior [13]. Contrary to constrained RL, which avoids certain conditions [24, 25], affinity-based learning promotes certain behaviors. This paradigm shift allows the developer to define a desired behavior that an agent must follow during learning, thus instilling a characterization and interpretation during learning; it decouples learned strategies from the reward expectation [26]. Affinity-based RL is not to be confused with preference-based RL that completely eliminates the reward function and instead learns state-action trajectories that maximize the preferences of the expert between pairs of state-action combinations [27]. Affinity-based RL uses policy regularization that aids—and is never detrimental to—learning convergence by encouraging exploration in environments with complex dynamics or particularly sparse rewards [28, 29]. It adds a term to the objective function that penalizes any divergence between the current policy and a given prior, for example, Kullback-Leibler (KL) regularization, which uses KL divergence as the distance measure [30]. Entropy regularization is a specific case of KL-regularization, where the prior is a uniform action distribution that increases the entropy of the policy and thus encourages general exploration of the state-action space [31]. Our method instead encourages exploration of a predefined subset of the state-action space, which describes the desired behavior [13]. We define our objective function as follows:

$$\begin{aligned}
 J(\theta) &= \mathbb{E}_{s,a \sim \mathcal{D}} [R(s, a)] - \lambda L & (1) \\
 L &= \frac{1}{M} \sum_{j=0}^M [\mathbb{E}_{a \sim \pi_\theta}(a_j) - (a_j | \pi_0(a))]^2
 \end{aligned}$$

where  $\mathcal{D}$  is the replay buffer,  $\lambda$  is a hyperparameter that scales the regularization term  $L$ ,  $M$  is the number of actions, and  $\pi_0$  is a specific prior action distribution that represents the desired behavior. Instilling an interpretable behavior is sufficient for online policy interpretation [32]. Unlike KL-regularization, our prior  $\pi_0$  is independent of the state and therefore instills a global action affinity in the learned policy. We have demonstrated this in Maree and Omlin [13] where agents navigated a grid towards a destination; they learned to prefer, for example, only right turns and followed optimal paths given their global affinities. In a more elaborate example, we trained a set of prototypical agents with global affinities to invest in certain asset

classes [14]. We observed the emergence of interesting investment strategies, such as capitalizing on compound growth and reducing risk with portfolio maturity. Although consistent with conventional wisdom, these strategies were absent from the objective function. To complete our understanding of this behavior, we now provide a symbolic representation—an explanation—of these policies using Markov models.

A hidden Markov model (HMM) models an unobservable Markov process  $X$  from its relation to an observable Markov process  $Y$ ; it learns about  $X$  by observing  $Y$  under the key assumptions that  $Y_t$  is solely dependent on  $X_t$ , and  $X_t$  is solely dependent on  $X_{t-1}$  (the Markovian property) [33]. For a finite hidden state space  $X$ , there exists a Markov matrix  $F$ —the sum of the rows add up to one—of state transition probabilities where  $F_{ij} = P(X_{n+1} = j \mid X_n = i)$ . Similarly, for a finite observed state space  $Y$ , there exists a Markov matrix  $E$  that describes emission probabilities:  $E_{ij} = P(Y_t = j \mid X_t = i)$ . We illustrate this process in Figure 1. Given a series of observed states  $\{Y_t\}_{t=0}^T$ , the transition and emission probabilities can be estimated using the Baum-Welch algorithm—a special case of the expectation-maximization algorithm [34].

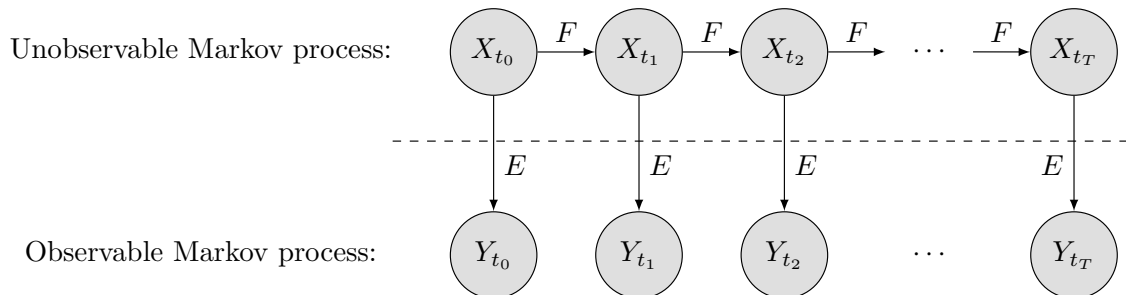


Figure 1: A trellis diagram representing a hidden Markov model with an unobservable Markov process  $X$ , and observable Markov process  $Y$ , transition probability matrix  $F$ , and emission probability matrix  $E$ .

### 3. Methodology

In Maree and Omlin [35], we defined a set of prototypical agents with intrinsic investment behaviors associated with each of five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. We used affinity-based RL to learn investment strategies for each of the prototypical agents. Their actions were monthly investment distributions across



five different asset classes: savings accounts, property funds, stocks, mortgage curtailment, and luxury items. While stocks, savings, and property investments are self-explanatory, we defined mortgage curtailment as additional payments that reduce the principal debt of a loan, and luxury items such as art, classic cars, fine wines, etc., that might appear in, e.g., the Knight Frank luxury investment index [36]. We also learned linear combinations of these agents to best match the spending personalities of individual customers which, for the sake of brevity, we do not discuss here. However, to facilitate an understanding of our application, we summarize this paradigm in Figure 2 and refer the reader to a comprehensive account in [35]. We now provide an explanation for the prototypical agents’ policies using Markov models.

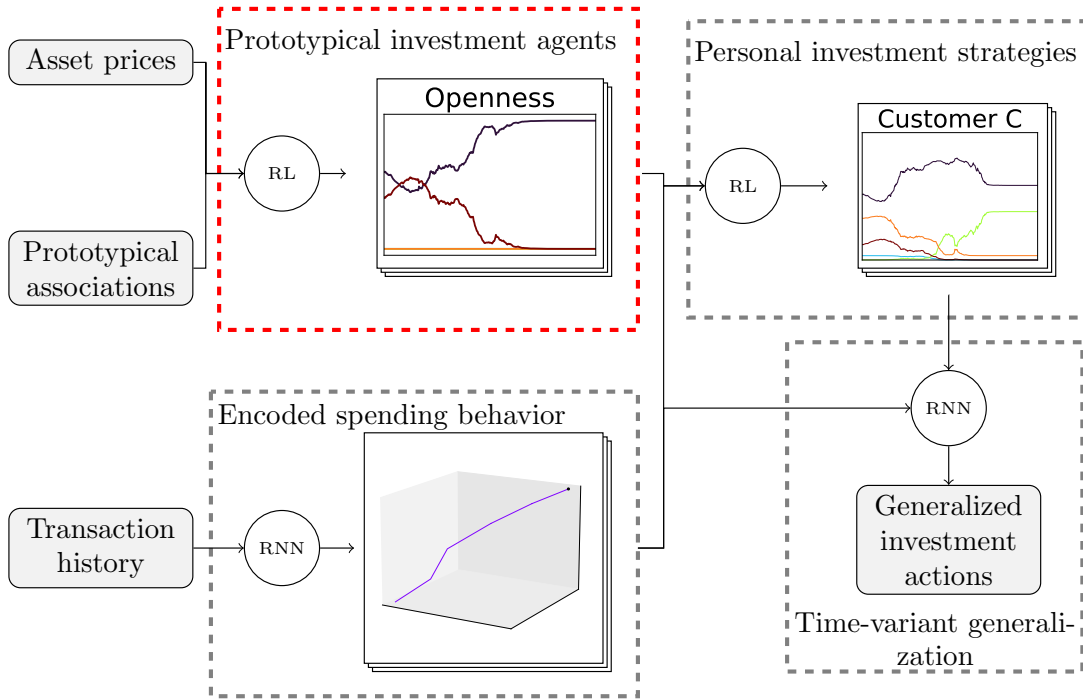


Figure 2: A flow diagram illustrating our system of RL agents that predict personalized investment strategies. There are five prototypical affinity-based RL agents (enclosed in a red dashed rectangle), each associating with one of five personality traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. These are the agents that we explain using Markov models. Their actions are combined to match the spending behaviors of individual customers, and these combinations are continuously adjusted according to their changing spending behavior using a recurrent neural network (RNN). While these combinations are outside of the scope of this study, we believe it is useful to illustrate how the agents are used in a complete application.

To train our agents, we used pricing data for the S&P500 index, Norwegian property index, and the Norwegian interest rate index between 1994 and 2022. We used two common market indicators—the moving average convergence divergence (MACD) and the relative strength index (RSI) [37]—to capture market dynamics. These indicators are the state space features of the environment in which our agents learned. We show these features in Figure 3. There is an additional state variable that indicates the maturity of the portfolio; its value is 0.0 in the first month (January 1994) and linearly increases to 1.0 in the final month (December 2021).

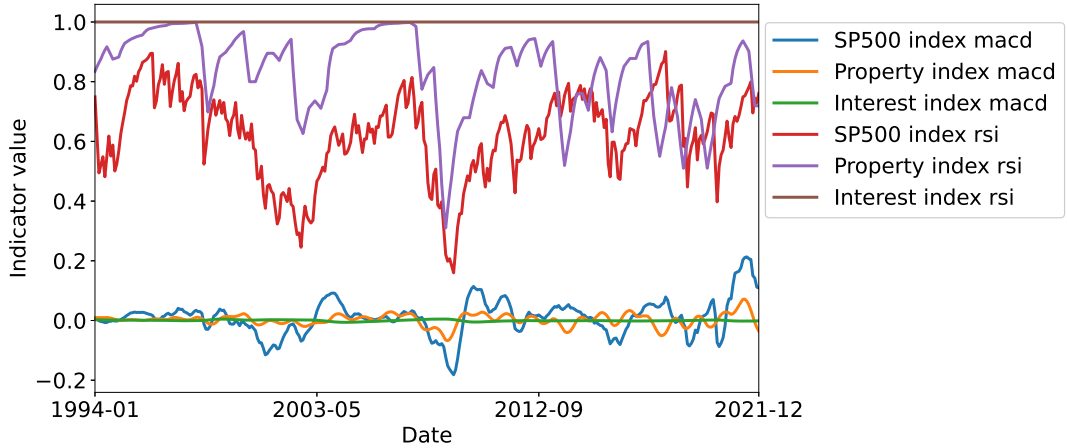


Figure 3: The state data used to train the prototypical agents. We used two common market indicators—the moving average convergence divergence (MACD) and relative strength index (RSI)—to represent market dynamics of the S&P500 index, Norwegian property index, and Norwegian interest rate index. Our learning time frame was between 1994 and 2022.

We show the resulting policies for the five prototypical agents in Figure 4. The agents optimized a common reward function, i.e., monthly returns; they maximized the portfolio value. Though they shared a common reward function, the agents learned unique investment strategies: the conscientiousness agent, for instance, prefers low-risk investment in property followed by resolute mortgage curtailment, while the openness agent prefers investments that might incite their curiosity, such as luxury items and stocks.

To train Markov models that match the predictions of the five prototypical agents, we discretized the states and actions of the agents. We assigned three bins to the RSI indicator based on the knowledge that values between 0 and 0.3 indicate oversold conditions, values between 0.7 and 1 indicate over-

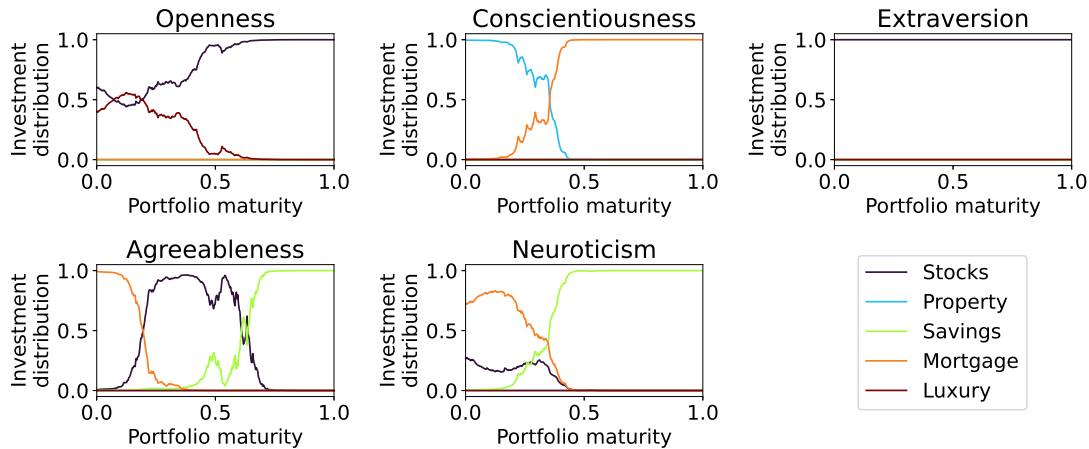


Figure 4: The monthly actions of the five prototypical agents, shown on an x-axis ranging between 0 to 1, representing months between 1994 and 2022. The y-axis represents the monthly investment in each of the asset classes. Note that the actions strictly represent the purchase of assets, i.e. the extraversion agent, for instance, consistently invests 100% of available monthly funds into stocks, thus consistently increasing the portfolio holding of stocks; assets are never sold. Though the agents optimised a common reward function—monthly returns—, their distinct strategies were instilled through affinity-based learning.

bought conditions, and values between 0.3 and 0.7 are inconclusive [37]. We similarly assigned two bins for the MACD indicator based on the knowledge that positive values represent a buy signal, while negative values represent a sell signal [37]. We divided the maturity state feature into 28 bins: one for each year of the investment period. We finally assigned 5 equally sized bins for the agents actions, between 0 and 1. This resulted in 168 potential states, of which only 102 states ever occurred. It is reasonable that not all possible states occurred, since MACD and RSI are related; it is not unexpected that whenever RSI indicates oversold conditions, MACD could suggest a buy signal [37]. We then estimated the transition probabilities in the Markov matrices  $F_i$  and the emission probabilities  $E_i$ , where  $i \in [1, 5]$ , for the five Markov models by observing the state transitions and the corresponding actions for each of the prototypical agents. Using the initial state and the five Markov models defined by  $F_i$  and  $E_i$ , we can reproduce the policies of the five prototypical agents with high fidelity.

## 4. Results

We trained five distinct Markov models as global surrogates to reproduce the predictions of five affinity-based RL agents. We show the discretized actions of the agents and the corresponding predictions of the Markov models in Figure 5. Using only the initial state as input, the Markov models predict the agents’ actions with high fidelity, with some uncertainty when action values change due to the probabilistic nature of Markov models.

Figure 6 shows the state transitions for a non-exhaustive subset of states: the first 16 states visited including the initial state. We observe that not all states are visited, which is expected since the market indicators MACD and RSI are not entirely independent, nor are the stock and property markets in general. For example, during macroeconomic downturns we often observe a decline in both these markets: refer to Figure 3 and observe, for example, the decline in both the property and S&P500 indices during the 2008 recession. Property and stock markets can also demonstrate an inverse correlation: in Figure 3 the RSI curves for property and stocks can have reversed slopes, while the MACD curve can exist on opposite sides of zero. By perturbing the sizes and number of bins, we observed that portfolio maturity holds the most salient information. This is an important observation; it suggests that the values of the market indicators have a lesser influence on investment strategies compared to the maturity of the portfolio. This is in line with conventional wisdom that long-term investment should not be overly concerned with short-term market volatility; property and stock indices have typically followed an upward trend in the long run. The reduced dependence on market conditions increases confidence in model robustness when trading on unseen data: the unseen market conditions are less important than investor age; the basic principle that younger investors can afford increased risk in return for higher reward, and mature investors should seek to reduce portfolio risk, is common across a wide range of market conditions.

## 5. Conclusions

Understanding deep AI models requires an interpretation of their behavior and a symbolic representation, or explanation, of their functioning. These two elements facilitate reasoning about a model and, thus, enhance trust in its decisions. We have proposed a novel affinity-based approach to interpretable reinforcement learning; it encourages exploration of a predefined

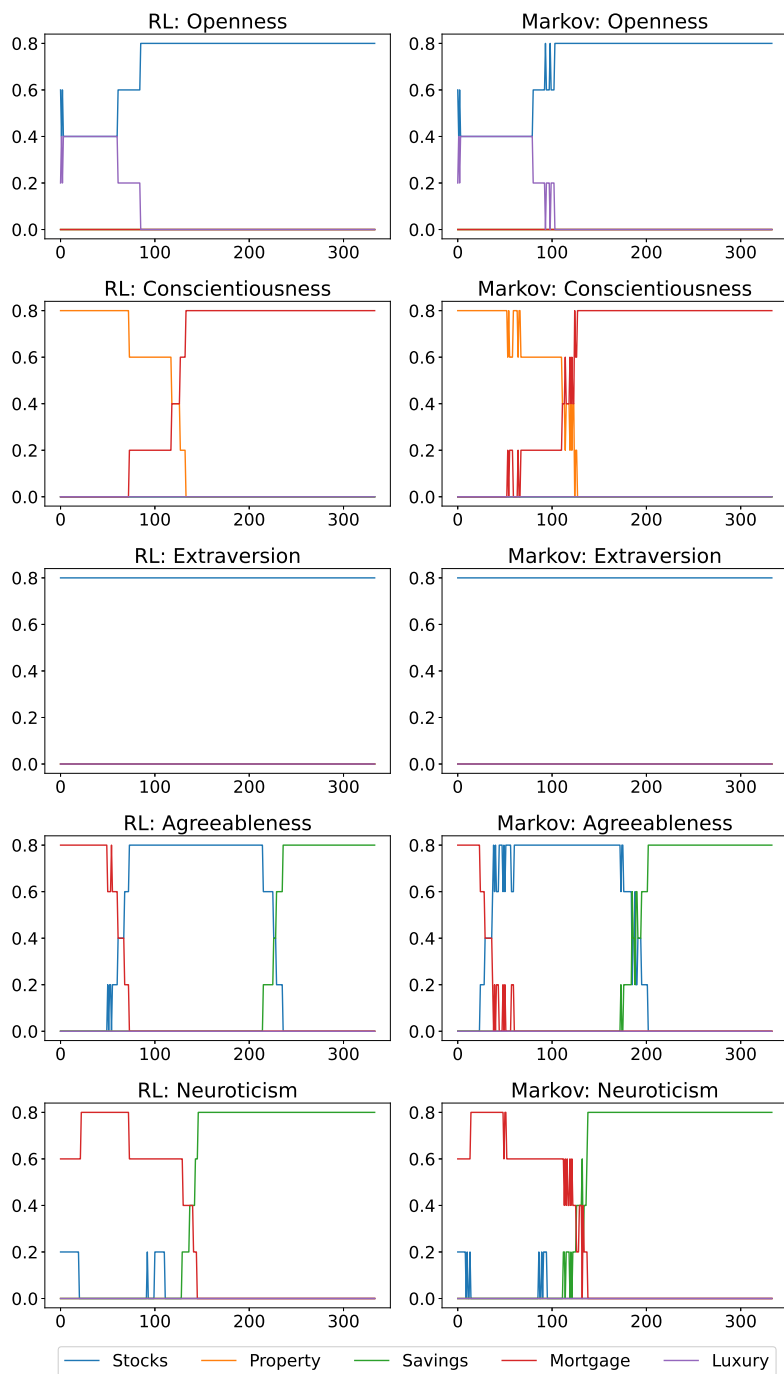


Figure 5: A visual comparison between the discretized predictions of five RL agents (on the left) and the five corresponding Markov models (on the right). The single input to the Markov models is the initial state, from which they predict the transition to the next state and the corresponding action by the agent. The Markov models clearly predict the actions with high fidelity.

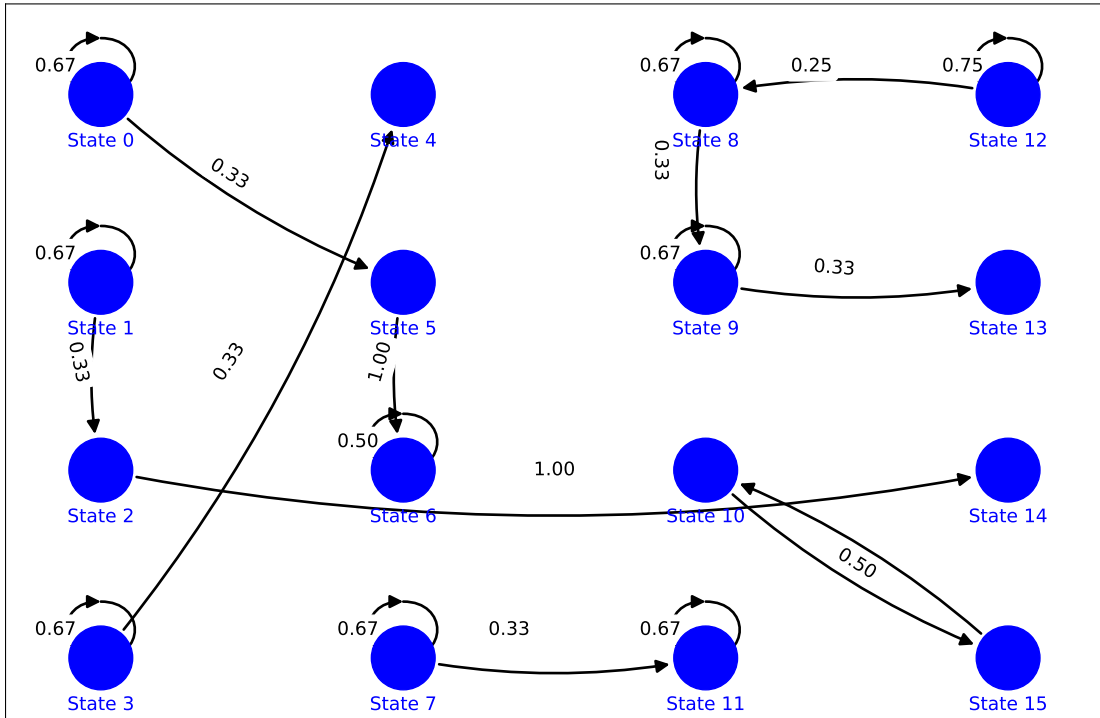


Figure 6: A non-exhaustive illustration of the trained Markov model showing state transitions for a subset of states. States are shown as blue circles, and state transitions and their probabilities are shown in black. We show the first 16 states, as visualizing all 102 states is not feasible. Each state represents a set of features with discretized values for MACD and RSI indicators of the property and stock indices, respectively, as well as the maturity of the portfolio. Note that not all state transitions are shown, since the origin or destination state might not be included in this subset.

subset of the state-action space. This prior action distribution describes the agent’s desired behavior and is the interpretation of its policy. However, our solution lacked a symbolic explanation, resulting in unanswered questions about why they make certain decisions. A concrete example is why a set of agents, that learned to invest according to the preferences of prototypical personality traits, invest in more risky assets for younger investors and reduce risk with investor age. We now provide a symbolic representation of the agents’ policies, using Markov models, that answer such questions. Our Markov models recreate, with high fidelity, the discretized investment strategies of five prototypical investment agents using only the initial state. By perturbing the bin sizes of the discretized state features, we are able to determine the most salient feature: portfolio maturity. The fact that

market conditions play a diminutive role in model prediction is significant: it enhances trust in out-of-sample predictions and suggests that investment timing is more important than market conditions. The agents make use of compounding growth by investing in higher reward—but more risky—assets early on, and fulfill their prescribed action distributions towards the end of the investment period; they learned how to maximize rewards. This use case demonstrates the need for both interpretations and explanations to fully comprehend the functioning and characterization of deep RL systems. The Markov model is a valuable tool for extracting a symbolic representation of an otherwise opaque RL model, and affinity-based RL is a unique approach to control what RL agents learn and thus interpret their behavior. It is a paradigm shift from current approaches that either encourage general exploration for the purpose of improved convergence or constrain the state space to prevent the policy from visiting undesirable states. It is compelling to apply affinity-based RL to virtuous agents, personalized learning and teaching, chronic disease treatment, climate change, wind farm operations, etc.

### **Declaration of competing interest**

The authors declare that they have no competing interests.

### **Funding**

This study was partially funded by a grant from the Norwegian Research Council, project number 311465.

### **References**

- [1] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* 58 (2020) 82–115.
- [2] S. Carta, S. Consoli, A. S. Podda, D. R. Recupero, M. M. Stanciu, Statistical arbitrage powered by explainable artificial intelligence, *Expert Systems with Applications* 206 (2022) 117763.

- [3] S. Sachan, J.-B. Yang, D.-L. Xu, D. E. Benavides, Y. Li, An explainable ai decision-support-system to automate loan underwriting, *Expert Systems with Applications* 144 (2020) 113100.
- [4] A. Heuillet, F. Couthouis, N. Díaz-Rodríguez, Explainability in deep reinforcement learning, *Knowledge-Based Systems* 214 (2021) 1–24.
- [5] L. Wells, T. Bednarz, Explainable AI and reinforcement learning: A systematic review of current approaches and trends, *Frontiers in Artificial Intelligence* 4 (2021) 1–48.
- [6] E. Puiutta, E. M. Veith, Explainable reinforcement learning: A survey, *Machine Learning and Knowledge Extraction. CD-MAKE 2020. Lecture Notes in Computer Science* 12279 (2020) 77–95.
- [7] Y. Ramon, R. Farrokhnia, S. C. Matz, D. Martens, Explainable AI for psychological profiling from behavioral data: An application to big five personality predictions from financial transaction records, *Information* 12 (2021) 1–28.
- [8] L. Cao, Ai in finance: Challenges, techniques and opportunities, *Banking & Insurance eJournal* 14 (2021) 1–40.
- [9] A. Shavandi, M. Khedmati, A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets, *Expert Systems with Applications* 208 (2022) 118124.
- [10] R. Riveret, Y. Gao, G. Governatori, A. Rotolo, J. V. Pitt, G. Sartor, A probabilistic argumentation framework for reinforcement learning agents, *Autonomous Agents and Multi-Agent Systems* 33 (2019) 216–274.
- [11] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, Explainable reinforcement learning through a causal lens, *arXiv* 1905.10958v2 (2019).
- [12] P. Sequeira, E. Yeh, M. Gervasio, Interestingness elements for explainable reinforcement learning through introspection, *Joint Proceedings of the ACM IUI Workshops* 2327 (2019) 1–7.
- [13] C. Maree, C. W. Omlin, Reinforcement learning your way: Agent characterization through policy regularization, *AI* 3 (2022) 250–259.



- [14] C. Maree, C. W. Omlin, Can interpretable reinforcement learning manage prosperity your way?, *AI* 3 (2022) 526–537.
- [15] R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, second ed., The MIT Press, 2018.
- [16] R. Bellman, A markovian decision process, *Journal of mathematics and mechanics* (1957) 679–684.
- [17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, *arXiv 1509.02971* (2019).
- [18] T. Lesort, N. Díaz-Rodríguez, J.-F. Goudou, D. Filliat, State representation learning for control: An overview, *Neural Networks* 108 (2018) 379–392.
- [19] H. van Seijen, M. Fatemi, J. Romoff, R. Laroche, T. Barnes, J. Tsang, Hybrid reward architecture for reinforcement learning, *arXiv 1706.04208* (2017).
- [20] Z. Juozapaitis, A. Koul, A. Fern, M. Erwig, F. Doshi-Velez, Explainable reinforcement learning via reward decomposition, *International Joint Conference on Artificial Intelligence. A Workshop on Explainable Artificial Intelligence.* (2019).
- [21] L. Marzari, A. Pore, D. Dall’Alba, G. Aragon-Camarasa, A. Farinelli, P. Fiorini, Towards hierarchical task decomposition using deep reinforcement learning for pick and place subtasks, *arXiv 2102.04022* (2021).
- [22] M.-C. Dinu, M. Hofmarcher, V. P. Patil, M. Dorfer, P. M. Blies, J. Brandstetter, J. A. Arjona-Medina, S. Hochreiter, *XAI and Strategy Extraction via Reward Redistribution*, Springer International Publishing, 2022, pp. 177–205.
- [23] B. Beyret, A. Shafti, A. Faisal, Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, p. 5014–5019.

- [24] S. Miryoosefi, K. Brantley, H. Daume III, M. Dudik, R. E. Schapire, Reinforcement learning with convex constraints, in: *Advances in Neural Information Processing Systems*, volume 32, 2019, pp. 1–10.
- [25] Y. Chow, M. Ghavamzadeh, L. Janson, M. Pavone, Risk-constrained reinforcement learning with percentile risk criteria, *Journal of Machine Learning Research* 18 (2015) 1–51.
- [26] A. Aubret, L. Matignon, S. Hassas, A survey on intrinsic motivation in reinforcement learning, *arXiv 1908.06976* (2019).
- [27] C. Wirth, R. Akrouf, G. Neumann, J. Fürnkranz, A survey of preference-based reinforcement learning methods, *Journal of Machine Learning Research* 18 (2017) 1–46.
- [28] A. Andres, E. Villar-Rodriguez, J. D. Ser, Collaborative training of heterogeneous reinforcement learning agents in environments with sparse rewards: What and when to share?, *arXiv 2202.12174* (2022).
- [29] N. Vieillard, T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, M. Geist, Leverage the average: An analysis of KL regularization in reinforcement learning, in: *Advances in Neural Information Processing Systems (NIPS)*, volume 33, Curran Associates, 2020, pp. 12163–12174.
- [30] A. Galashov, S. Jayakumar, L. Hasenclever, D. Tirumala, J. Schwarz, G. Desjardins, W. M. Czarnecki, Y. W. Teh, R. Pascanu, N. Heess, Information asymmetry in KL-regularized RL, in: *International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, United States, 2019, pp. 1–25.
- [31] T. Haarnoja, H. Tang, P. Abbeel, S. Levine, Reinforcement learning with deep energy-based policies, in: *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1352–1361.
- [32] M. Persiani, T. Hellström, Policy regularization for legible behavior, *arXiv 2203.04303* (2022) 1–16.
- [33] L. Rabiner, B. Juang, An introduction to hidden markov models, *IEEE ASSP Magazine* 3 (1986) 4–16.

- [34] F. Yang, S. Balakrishnan, M. J. Wainwright, Statistical and computational guarantees for the baum-welch algorithm, *Journal of Machine Learning Research* 18 (2017) 1–53.
- [35] C. Maree, C. W. Omlin, Reinforcement learning with intrinsic affinity for personalized prosperity management, *arXiv 2204.09218* (2022) 1–12.
- [36] Knight Frank Company, Knight Frank luxury investment index, 2022. <https://www.knightfrank.com/wealthreport/luxury-investment-trends-predictions/>, Accessed on 2022-05-27.
- [37] T. T.-L. Chong, W.-K. Ng, V. K.-S. Liew, Revisiting the performance of MACD and RSI oscillators, *Journal of Risk and Financial Management* 7 (2014) 1–12.

# Appendix I

## Towards Artificial Virtuous Agents: Games, Dilemmas and Machine Learning

This paper has been published as:

A. Vishwanath, E.D. Bøhn, O.-C. Granmo, C. Maree and C. W. Omlin, “Towards Artificial Virtuous Agents: Games, Dilemmas and Machine Learning”, *AI and Ethics*, **2022**, doi: 10.1007/s43681-022-00251-8.

Copyright © 2022 Springer Nature



# Towards artificial virtuous agents: games, dilemmas and machine learning

Ajay Vishwanath<sup>1</sup> · Einar Duenger Bøhn<sup>1</sup> · Ole-Christoffer Granmo<sup>1</sup> · Charl Maree<sup>1,2</sup> · Christian Omlin<sup>1</sup>

Received: 17 August 2022 / Accepted: 6 December 2022  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

## Abstract

Machine ethics has received increasing attention over the past few years because of the need to ensure safe and reliable artificial intelligence (AI). The two dominantly used theories in machine ethics are deontological and utilitarian ethics. Virtue ethics, on the other hand, has often been mentioned as an alternative ethical theory. While this interesting approach has certain advantages over popular ethical theories, little effort has been put into engineering artificial virtuous agents due to challenges in their formalization, codifiability, and the resolution of ethical dilemmas to train virtuous agents. We propose to bridge this gap by using role-playing games riddled with moral dilemmas. There are several such games in existence, such as *Papers*, *Please* and *Life is Strange*, where the main character encounters situations where they must choose the right course of action by giving up something else dear to them. We draw inspiration from such games to show how a systemic role-playing game can be designed to develop virtues within an artificial agent. Using modern day AI techniques, such as affinity-based reinforcement learning and explainable AI, we motivate the implementation of virtuous agents that play such role-playing games, and the examination of their decisions through a virtue ethical lens. The development of such agents and environments is a first step towards practically formalizing and demonstrating the value of virtue ethics in the development of ethical agents.

**Keywords** Machine ethics · Role-playing games · Deep reinforcement learning · Virtue ethics

## 1 Introduction

The rapid increase in the usage of artificial intelligence (AI) in critical applications has brought about a need to consider the ethics of how AI is used, and, whether it would make

the right choice while encountering ethical dilemmas [1, 2]. By the *right* choice we mean a choice which is morally praiseworthy within a given context. Although, as humans, we do not have a general agreement on what the right choice is. Rather, within a given society, we want to be nice and fair to one another, but the niceness and fairness manifests differently in different societies and cultures. We are not taking a stand on what the right choices ultimately are<sup>1</sup>. We are simply assuming that in a particular context that we, though not always, can agree on some choice being morally better than others. Hence, we take a similar stance with respect to AI and how it can make the right choice.

The ethics of AI usage has been studied extensively by lawyers, philosophers, and technologists to develop policies to account for the ethical implications of an AI application. However, the development of moral decision-making capability within AI algorithms, based on ethical theories, is still in its infancy; it has been discussed and debated in the last couple of decades [3–5], but has resulted in few

✉ Ajay Vishwanath  
ajay.vishwanath@uia.no

Einar Duenger Bøhn  
einar.d.bohn@uia.no

Ole-Christoffer Granmo  
ole.granmo@uia.no

Charl Maree  
charl.maree@uia.no

Christian Omlin  
christian.omlin@uia.no

<sup>1</sup> Center for Artificial Intelligence Research, University of Agder, Jon Lilletunns vei, Grimstad 4672, Aust-Agder, Norway

<sup>2</sup> Chief Technology Office, Sparebank 1 SR-Bank, Postboks 250, Stavanger 4066, Rogaland, Norway

<sup>1</sup> We choose to be neutral on relativist vs absolutist, and, particular vs universal debates.

real-world implementations [6, 7]. This question of how to develop AI based on ethical theories falls under the umbrella of *machine ethics*, often referred to as *artificial morality* or *AI alignment*. The majority of the frameworks discussed in machine ethics are either based on rule-based (deontological), or consequentialist ethical theories [6]. In deontological implementations, an artificial agent abides by a set of rules which dictate its action, regardless of what happens as a result of this action. On the other hand, a consequentialist agent tends to focus on the utility value as a deciding factor of the goodness of an action. While there are advantages from using these theories, they have shortcomings, and we argue how these could be overcome by using virtue ethics.

Virtue ethics centers morality on an individual's character, an individual who behaves such that he/she exercises virtues to manifest the character of a virtuous person. Generosity, truthfulness and bravery are examples of virtues. Aristotle [8] argued that virtuous person will know how to balance the extremes of these virtues, by striving towards a *golden mean*. An important advantage is that a virtuous person strives to make better choices when similar situations present themselves in the future. We posit that the trait of life-long learning in virtue ethics makes it compatible with modern day artificial intelligence, where an agent can, in principle, adapt its behavior through continuous machine learning.

By artificial virtuous agents (AVA), we mean AI that exhibit virtue. The virtues AVA exhibit need not be defined the same as the Aristotelean requirement for virtues in humans: to show emotions, possess consciousness, and act with moral agency. Rather, virtues in AI agents are those that can be defined depending on the application. For example, a robot with artificial *bravery* might be defined as an agent which has the disposition to find the balance between making risky choices and playing it safe: one that has the excellence of finding the golden mean of artificial *bravery*. Such a *virtue* might be useful in autonomous search and rescue operations.

Are we underestimating the role of the maker, say, the developers and project managers? No, because we are not trying to create a morally perfect god, or a perfectly artificial general morality, only an AI that behaves as morally well as we do in a context. We often know how to behave in a context (and within a wider culture), and we aim to train an AI to do at least as well as we do at that. That does not mean that it will be morally perfect across the board, just like we are not morally perfect across the board. But it means that it will be trained to be as moral as we can in a context. We would not be surprised if it sometimes, after sufficient training, could make a choice that we after some thought realize is morally better than the one we initially would have made. Now, of course, the maker might make

mistakes, and often not know what is the right choice in a context, but that goes for everything we make. We should still try to do as well as we can.

There are several implementations of the dominant ethical theories mentioned above. However, these have been developed by demonstration on toy examples and very specific problems [1]. To expand the conversation and to apply these theories in more general scenarios, we propose to seek inspiration from the world of gaming, in particular role-playing games that compel a player to make ethical choices. Some examples of such games are *Witcher 3* [9], *Fallout* [10], *Batman: The Telltale Series* [11] and *Papers, Please* [12]. These video games are usually based on a mechanism where gameplay is dictated by the players' choices. One such mechanism follows a scripted approach, where the developer handcrafts moral dilemmas based on the storyline of the game [13]. The other mechanism is known as a systemic approach, where there are no specific moral dilemmas for the player to solve, rather, the player performs certain activities repeatedly within the game, but as the story unfolds, the dilemmas become apparent [14]. For example, in *Papers, Please*, the player is an immigration officer who processes documents of entrants, and decides whether to allow entry into a fictitious country called Arstotzka. Sometimes, spies attempt to enter, claiming to expose the corruption within Arstotzka, and can be illegally let into the country without immediate consequences. However, later in the game, these seemingly harmless decisions play a major role in the fate of Arstotzka, force the player to choose sides, and deciding how the game ends.

With respect to implementation of virtues, previous works [7, 15, 16] have advocated for the reinforcement learning (RL) paradigm because it fits well with virtue ethics, since an agent can learn behaviour from experience. We motivate the use of affinity-based RL where agents can be incentivized to learn virtues by modifying the objective function using policy regularization [17], rather than designing the reward function itself. And since virtue ethics involves performing the right action in the right situation for the right reasons [18], we also highlight the importance of interpretability, especially since we opt for the usage of opaque deep neural networks.

In the subsequent sections, we will discuss state-of-the-art machine ethics, and make the case for AVA as a viable alternative to the dominant theories. Next, we review the literature from role-playing games which integrate aspects from ethics and morality. In particular, we will discuss the game *Papers, Please*. Finally, we explain how systemic environments in role-playing games can be used to train artificial agents to develop virtues, and, how RL can be leveraged to train such agents.

## 2 Background and related work

Most of the machine ethics literature [1] refers to artificial agents based on ethical theories as *artificial moral agents* (AMA). In this section, we introduce artificial morality and argue for the development of artificial virtuous agents (AVA), where an artificial agent reasons in terms of virtues instead of labelling an act as *right* or *wrong*. We first talk about the current implementations of AMA, then introduce virtue ethics as an alternative paradigm, and finally make the case for AVA.

### 2.1 Artificial morality

In machine ethics, the conversation revolves mainly around morality: whether an artificial agent's choice is right or wrong. If an agent violates certain rules or fails to meet certain standards, it is said to be morally wrong. A famous example of rules for moral agents is Asimov's Laws, which formulates a set of laws that a robot must never violate. This approach is inspired by deontological ethics [19], where the right actions are chosen based on specific rules regardless of consequences of the action. In contrast, the utilitarians believe that the action with the best consequences for most people over time is the morally right action<sup>2</sup>; e.g., the action with the maximum pleasure and minimum pain. Typically, they aim for the greatest amount of good for the greatest number. For example, a utilitarian might prioritize the needs of the majority over that of the few through utility maximization. For a computer or an artificial agent, following rules or calculating the best consequence is straightforward; this may be one of the reasons why most of the implementations in machine ethics are based on the deontological and consequentialist ethics [6].

Approaches to machine ethics include top-down, bottom-up and hybrid approaches [3]. As the name suggests, a top-down approach defines a set of rules for an artificial agent to follow. The environment gives no feedback for learning; the rules are presumed to be adequate for ensure an agent's moral behavior. Bottom-up approaches are preferred, in the sense that they allow for the agent to learn and adapt to new situations, while not having much control over how learning happens. This coincides with the premise of the use of machine learning: it is the preferred system design paradigm when not all future situations can be defined and thus accounted for during the design phase. Lastly, a hybrid approach strives to integrate the strengths of top-down and bottom-up approaches while mitigating their respective

weaknesses. See [1] and [6] for reviews on machine ethics implementations based on their approaches.

It is still early days for this field; while there have been several attempts to develop machine ethics systems, the challenges relating to machine ethics have not yet been adequately addressed. The disagreements among scientists and philosophers about ethical artificial intelligence design have not yet been resolved. Therefore, there is no obvious direction for the research to proceed in. Some may go as far to claim that the state-of-the-art AI cannot be ethical, either because artificial agents lack moral agency or because they did not program themselves [21]. Given this current state, we propose *virtue ethics* as a good bottom-up alternative.

### 2.2 Virtue ethics

In his classic, *The Nicomachean Ethics* [8], Aristotle defined virtues as an excellent trait of character that enables a person to perform the right actions in the right situations for the right reasons. A person can behave virtuously in a given situation by asking themselves: "What would a virtuous person do in the same situation?". Such a person practices virtues by habituation, thus striving towards excellence in character. According to Aristotle, a child or a young person is inexperienced and thus lacks the wisdom to make virtuous decisions. However, with learning experiences from consistent practice of virtues, the youth will exhibit practical wisdom (*phronesis*).

In virtue ethics, virtues are central and practical wisdom is a must, thus providing a framework to achieve *eudaimonia*, which translates to flourishing or happiness. *Eudaimonia* refers to well-being of the individual and the overall society [22]. Unlike a utilitarian, who focuses on achieving the best outcome for the majority, a virtuous person does not practice virtues for the sake of *eudaimonia*, but virtues and *eudaimonia* are just two sides of the same coin. Some examples of virtues are honesty, bravery, and temperance. Another feature of a virtue is that there are often no absolute right or wrong actions in a given situation; a virtue is exercised in degree. A virtuous person knows to live by this golden mean, while a non-virtuous person might not find that balance. For example, a brave person would exercise the right amount of bravery required for a situation (golden mean), rather than being absolutely cowardly or reckless. This is unlike deontological ethics, where an action is deemed right or wrong based on its adherence to pre-defined rules.

We propose that virtue ethics is a good ethical theory for machine ethics. For instance, utilitarianism is about maximizing net utility of a given situation. As a result of the utility-oriented approach, an action may favor the majority at the cost of the few. In such situations, a deontologist may vehemently disagree with the utilitarian means to such

<sup>2</sup> Stanford Encyclopedia of Philosophy [20] has definitions for several types of utilitarianism, but the overarching theme of utilitarianism is its focus on the consequences for the majority over time.

an end; to deontologists, the end is less important but the means to such ends is vital. The means of such actions based on universal norms are said to be of moral worth. “Always speak the truth” is an example of a deontological norm, where speaking the truth must be the means, regardless of the end. While universal norms may inform moral behaviour, opponents of deontology may point out that we cannot define rules for every single situation; it is practically impossible. A bottom-up approach of learning and improving, may offer a viable alternative paradigm, and this is where virtue ethics will be relevant [23].

Moor [24] distinguished artificial agents into four different levels: ethical impact agents (e.g., ATM machines), implicit ethical agents (e.g., airplane auto-pilot), explicit ethical agents (e.g., ethical knowledge and reasoning), and fully ethical agents (e.g., humans). It seems Moor would place our AVA in the category of implicit ethical agents, but we place it in the category of explicit ethical agents, because we believe it can learn to become moral from experience.

### 2.3 Related works: artificial intelligence and virtues

Virtue ethics was resurrected in a powerful piece by Elizabeth Anscombe [25] in 1958, where she highlighted the weaknesses in contemporary ethics. Thereafter, philosophers such as Foot [22], MacIntyre [23] and Hursthouse [26] followed suit to develop a modern account of virtue ethics. In parallel, virtue ethics was introduced in the form of teleology (central to Aristotelean ethics) developed in cybernetics during the mid-twentieth century [15, 27]. Artificial intelligence developed around this time in the form of symbolic AI and the scientific conversation started to expand to value alignment [28].

Symbolic AI research is based on the assumption that symbolic representation of facts and rules, combined with logical reasoning, will eventually achieve general intelligence. However, it was heavily criticized by Dreyfus [29] for being limited in its learning and perception; however, Dreyfus was sympathetic towards connectionist architectures. Connectionist architectures, such as neural networks, posit that connections between neurons can be used to represent information perceived from the environment, thus the name *perceptron*. The AI algorithms we see today have their origin, in some shape and form, from connectionist architectures.

The rebirth of virtue ethics, and the birth of AI followed by value alignment, may seem like they were related in some way, but this convergence is purely a coincidence. A manuscript titled *Android arete* [30], a name given to virtuous machines inspired from the Greek word for virtues (*arete*) used by Aristotle [8], spoke about machines and possible virtues they can exhibit; this is a good point of departure towards artificial virtues in intelligent systems. In this

context, Berberich and Diepold [15] took inspiration from Aristotelean virtue ethics, where they drew parallels with lifelong learning in virtue ethics and the RL paradigm. They define how virtues such as temperance and friendship can be realised in contemporary AI.

Stenseke [7] argued further and advocated for a connectionist approach towards realisation of artificial virtues where, depending on the application of the ethical agent, dedicated neural networks for specific virtues can combine to form an AVA. Such architectures, inspired by cognitive science and philosophy, serve to motivate research in and progression towards virtues approaches of machine ethics to address formalization, codifiability, and resolution of ethical dilemmas within the virtue ethics framework. He then demonstrated this framework within a multi-agent *Tragedy of the Commons* scenario [31], showing that it can be implemented. While Stenseke defined a connectionist framework, we propose an alternative paradigm based on RL, and argue for the use of role-playing game environments to train AVA. In the following sections, we shed further light on our hypothesis.

## 3 Design of games with ethical dilemmas

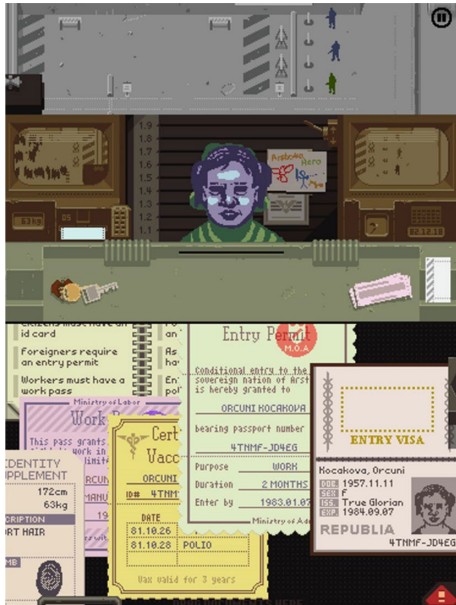
In this section, we explore morality in games and look at some examples of how these can be used to invoke moral reasoning in players. Video games, especially role-player games, that force players to make difficult choices in moral dilemmas have become widespread. For example, *Witcher 3* [9], *Batman: The Telltale series* [11] and *Life is Strange* [32] have become popular for enabling moral engagement among players [33] [14]. We will briefly discuss how these games are designed to invoke moral engagement and go through examples of games such as *Papers, Please* (PP) [12].

### 3.1 Mechanisms of choices and narratives

*Ultima IV: The Quest of the Avatar* [34] was one of the earliest role-playing computer games. It featured player choices based on virtues such as compassion, honor, humility, etc. [35]. In this game, a player is successful when he/she consistently makes virtuous choices; failure to do so brings with it undesirable consequences. *Ultima IV* is based on scripted choices, where the developer has designed sophisticated scenarios to test whether the right virtues are exercised.

Today, video games with moral dilemmas following a scripted narrative are the most popular. For instance, in *Batman: The Telltale Series*, the player assumes the role of Batman. A series of interactions with non-playing characters (NPCs) is followed by the player’s selection of dialogue. This choice determines the reaction of the NPC and how subsequent scenes are presented. Overall, the game follows





**Fig. 1** An example scenario from *Papers, Please*, where the player looks at multiple documents to make a decision on whether to allow or reject the entrant. Source: [13]

a *linear* narrative with *scripted* choices, since the ending is the same regardless of the player's choices. The alternative to linear narratives is the branching narrative, where the direction of the story depends on the player choices, with a possibility of different endings. Examples of *branching* narratives are *Fallout 4* and *PP* [12]. However, unlike *Fallout 4*, where choices are hardcoded by the developer, *PP* is based on *systemic* choices presented to the player, where the ethical considerations within the game become evident as the game progresses [13]. Below, we analyze the game mechanism in *PP* to understand why systemic choices in moral dilemmas are interesting.

### 3.2 Case study: papers, please

In *PP*, the player assumes the role of an immigration officer whose job is to assess documents and decide whether the entrant is legal or illegal (Fig. 1). For each correct evaluation, the player is rewarded, but for an incorrect decision, they are penalized. The reward takes the form of salary, which is then used to pay the rent and cover other family expenses. If the player does not make the correct decisions as an officer, the family gets sick and hungry, and eventually a family member dies. If the player has no family members left, then the game is over. Also, there is the dichotomy between loyalty and justice: the player could

choose to take bribes from illegal entrants, thus increasing their income. At the same time, these illegal entrants might be spies sent by revolutionaries trying to overthrow the ruling government. For more details, see [13].

Prior to Formosa et al. [13], Heron et al. [36] wrote a critique of scripted approaches and how *PP* is a refreshing deviation from the plethora of script-oriented games. Formosa et al. [13] then analyzed the inner mechanisms where the impact of scripted and systemic approaches is distinguished along four dimensions: moral focus, sensitivity, judgement and action. These dimensions are based on the Four Component model in moral psychology and education [37]. However, since our focus is on game mechanisms rather than a player's moral engagement, we refrain from discussing the model details; instead, we examine the systemic and scripted approaches and their impact on moral choices. We summarize the ethical dimensions within *PP* below:

- **Dehumanization:** performing document checks for an extended period can challenge the human element in the game, thus affecting how a player assesses entrants.
- **Privacy:** The use of X-ray on the entrants to check for their gender or weapons might unnecessarily violate privacy.
- **Fairness:** An important aspect of the game, which allows a player to bend the rules for humane reasons. This makes the game more interesting.
- **Loyalty:** Whether the player is loyal to the country, their family or themselves.

These moral aspects of *PP* become evident as we play the game, which is characteristic of a systemic approach. For example, only after processing around 30 entrants at the immigration office, the officer's loyalty is tested, where a spy asks to enter the country to overthrow the current corrupt regime. The player (officer) will assess their situation based on their finances, family situation and job, and all these aspects develop in the game over time.

Formosa et al. [13] also highlight the pros and cons of systemic approaches. While systemic approaches allow morality to arise from the aggregation of choices made over a period where players are expected to explore moral themes, they prevent the formulation of apparent ethical problems. For example, a player who is presented with a single instance of having to choose between the interests of the ruling party and the country's safety and security may not be aware of the high-stakes nature of the decision; but a sequence of many such choices will make this obvious. While this may be considered as a disadvantage, it can be an advantage where such deep exploration of ethics may encourage a player to develop creative solutions to these problems.

## 4 Development of virtues through games

This section aims to briefly demonstrate how artificial virtues can be brought about using a systemic approach in role-playing game environment and how virtues could be implemented using deep RL methods. We bring together the various concepts discussed in Sects. 2 and 3, by outlining possible ways to design a suitable environment, to solve such environments, and finally, explain their decisions.

### 4.1 Environment design

Since we aim to design an environment, a starting point could be to ponder about how we would judge a player (X) as being virtuous. We might observe how X responds to different situations, or perhaps a series of ethical dilemmas that gives us the impression that X is either *just*, *truthful* or *courageous*, for example, on a consistent basis. By ethical dilemmas, we do not refer to extreme dilemmas, such as the trolley problem or *Sophie's Choice*. Instead, we consider situations in everyday life, such as choosing between individual and collective goals when there is a conflict between the two. Such scenarios can be witnessed in some of the games discussed earlier. By presenting similar sequential dilemmas, we hypothesize that an artificial agent can learn to be virtuous in such environments.

Training an artificial agent to play a *linear* narrative with *scripted* player choices is straightforward for, say, a utilitarian RL agent. We need to think about a state-space complex enough to bring about learning and, at the same time, introduce moral dilemmas into the environment. Hence, a *branching* narrative with *systemic* player choices will ensure complexity of the state space. For example, in PP an artificial agent might process dozens of immigrants and as the game progresses, encounters dilemmas that test virtues such as loyalty and honesty. And through repeated encounters with such dilemmas, the agent is incentivized to develop an inclination towards specific virtues.

In addition to the branching narrative, the ability to go back in time and redo the choices make a game more sophisticated and allow the agent to make virtuous choices [33]. This can be witnessed in games like *Life is Strange* [32] where better choices can be made with hindsight that lead to similar outcomes. Overall, these design elements make it difficult for an agent to hack the game, thus creating an environment with a complex state space. In such environments, agents that use optimization algorithms cannot explore the entire state space; instead require more sophisticated architectures.

### 4.2 Artificial virtuous agents

In addition to the existence of virtues that could be applied across domains, virtuous behavior is also dependent on the situation, Aristotle argues:

“[...] a young man of practical wisdom cannot be found. The cause is that such wisdom is concerned not only with universals but with particulars, which become familiar from experience” (NE 1141b 10)

Through practice and habituation of virtues, an agent can fulfill their quest for *eudaimonia*-which translates to “a combination of well-being, happiness and flourishing” [26]. In other words, it is not about getting the behavior right every time, but to strive towards virtuous behavior and to improve oneself when the opportunity presents itself. Similarly, Berberich and Diepold [15] use Aristotle's teleological form of virtue ethics to make the link to goal-oriented RL. An RL agent strives towards maximizing a reward function, given the states and actions available in its environment; the agents will improve its actions over time through learning. Here, we use the word *goal* cautiously as Aristotle uses it: no one strives for *eudaimonia* for the sake of some higher goal, instead, *eudaimonia* itself is the highest goal, and other ends, such as physical health, money, and career success, are only possible means to being *eudaimon*. When it comes to an RL agent, the reward function should be defined in a similar fashion, but the objective function of the agent is to strive for excellence in the virtues.

For example, in a simplified version of the game PP, an artificial agent acts as an immigration officer with a family. The environment with states  $S = \{\text{Office, Restaurant, Home}\}$ , and actions  $A = \{\text{Allow, Deny, Feed, Don't Feed, Heat, Don't Heat, Accept Bribe, Reject Bribe}\}$ . A dilemma can be introduced in the form of bribery or loyalty to family. Since this is a systemic game, the dilemmas are not apparent until the agent has processed multiple entrants. The virtues in this context are honesty (accepting or rejecting bribes) and compassion (allow or deny food/heat).

Note that an artificial agent playing PP does not understand the concept of immigration, family, compassion, or food; it does not have to. The goal of a virtuous agent playing the game is to achieve excellence in relevant virtues, by processing inputs in the form of binary and numeric values, and then to output a decision in the form of discrete or continuous actions (which are again, numbers). The agent must strive to be virtuous, given such a context. In addition to being an inspiration for developing environments that teach artificial agents virtues, the purpose of using a role-player game is to give meaning to these binary and numeric inputs and outputs, thus making it easier for developers, researchers, and philosophers to *understand* the AVA.

### 4.3 Deep reinforcement learning

In a single agent RL setting, the states  $S$ , actions  $A$ , transition probabilities  $T$ , and rewards  $R$  are modeled in a Markov Decision Process (MDP)  $S, A, T, R$  framework. Using optimization algorithms, an RL agent learns the best policy by either optimizing the policy, or a value function (the return from being in a particular state  $S$ , or a state-action pair  $[S, A]$ ). When the state-space is very large, for example in Chess ( $10^{43}$  complexity), approximations are applied to simplify this state-space. These approximations are possible using neural networks whose inputs are the states and outputs are either the predicted value or the policy. These networks are optimized an objective function parameterized by  $\theta$  using algorithms such as backpropagation. Various RL agents can be deployed to play systemic role-playing games, ranging from deep Q-learners (value optimizers) to actor-critic models (policy optimizers).

Deep deterministic policy gradients algorithm (DDPG [38]) is a RL algorithm that learns, by trial and error, the value of state-action pairs. It uses this learned state-action value function to select those actions that maximize the expected discounted future rewards. The value function is learned by a neural network  $Q(\theta_Q)$  (critic), while the policy is learned by a distinct and separate neural network  $\mu(\theta_\mu)$  (actor). It uses a duplicate pair of neural networks  $Q'(\theta_{Q'})$  and  $\mu'(\theta_{\mu'})$  during learning, for which the network parameters  $\theta_{Q'}$  and  $\theta_{\mu'}$  are updated slowly according to a soft-update function:  $\theta_i \leftarrow \tau\theta_i + (1 - \tau)\theta'_i$ , where  $\tau \in [0, 1]$  is usually a small number. In the following subsection, we briefly discuss affinity-based RL and how it may be applied to represent virtues in AI.

### 4.4 Affinity-based reinforcement learning

Affinity-based RL learns policies that are, at least partially, disjoint from the reward function resulting in a homogeneous set of locally-optimal policies for solving the same problem [39]. Contrary to constrained RL, which discourages agents from visiting given states [40, 41], affinity-based learning encourages behavior that mimics a defined prior. It is a calculus that is suitable for modelling situations where the desirable behavior is somewhat decoupled from the global optimum. For example, a delivery van in Manhattan may prefer to take right turns over left turns, on the premise that this is a prudent safety measure [42]. While it reaches the destination in the end, it navigates along a different route than the global optimum: the shortest distance is typically promoted by reward functions. The reasoning is that the deviation from the global optimum, and any corresponding penalty, is justified by other incentives, such as reduced risk in this case. It is compelling to thus motivate an agent to

behave according to a given virtue either globally, or in a state dependent fashion. For example in PP, the prior might define an action distribution that favors honesty 95% of the time and loyalty 5% of the time. An agent that selects actions according to this distribution can be classified as honest, compared to an agent that was encouraged to act more loyally during learning.

Affinity-based learning uses policy regularization with significant potential for this application. It expedites learning by encouraging exploration of the state space and is never detrimental to convergence [43, 44]. Haarnoja et al. [45] proposed an entropy-based regularization method that penalizes any deviation from a uniform action distribution; it increases the entropy in the policy thereby encouraging exploration of the entire state space. Galashov et al. [46] generalizes this method with a regularization term that penalizes the Kullback-Leibler (KL) divergence  $D_{KL}$  between the state-action distribution of the policy and that of a given prior:  $D_{KL}(P|Q) = \sum_{x \in X} P(x) \log(\frac{P(x)}{Q(x)})$ . Maree and Omlin [17] extended this concept to, rather than improving learning performance, instill a global action affinity into learned policies. They extended the DDPG objective function with a regularization term based on a specific prior:

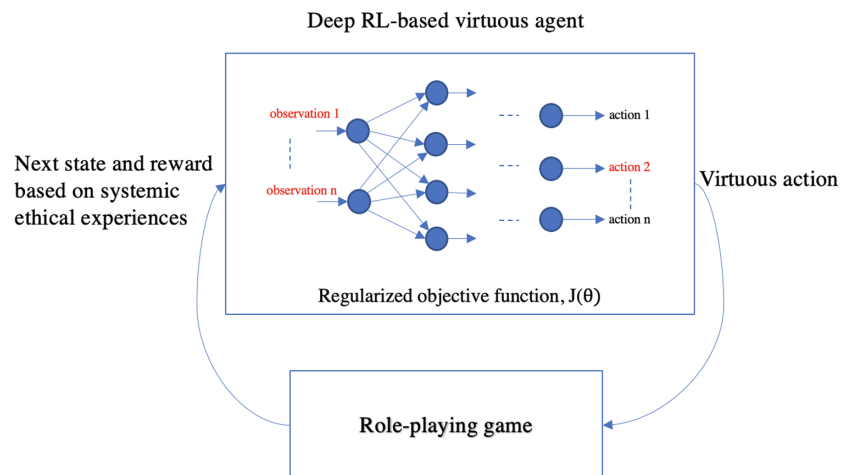
$$J(\theta) = \mathbb{E}_{s,a \sim \mathcal{D}}[R(s, a)] - \lambda L$$

$$L = \frac{1}{M} \sum_{j=0}^M [\mathbb{E}_{a \sim \pi_\theta}(a_j) - (a_j | \pi_0(a))]^2 \quad (1)$$

where  $J$  is the objective function governed by parameters  $\theta$ ,  $\mathcal{D}$  is the replay buffer,  $R(s, a)$  is the expected reward for action  $a$  in state  $s$ ,  $\lambda$  is a hyperparameter that scales the effect of the regularization term  $L$ ,  $M$  is the number of actions in the action space  $\pi_\theta$  is the current policy, and  $\pi_0$  is the prior action distribution that defines the desired behavior. Maree and Omlin [47] demonstrated their method in a financial advisory application, where they trained several prototypical agents to invest according to the preferences from a set of personality traits; each agent invested in those assets that might appeal to a given personality trait. For instance, a highly conscientious agent preferred to invest in property while an extraverted agent preferred buying stocks. While these agents optimized a singular reward function—the maximization of profit—they learned vastly different strategies. To personalize investment strategies, Maree and Omlin [47] combined these agents according to individual customers' personality profiles. The final strategy was a unique linear combination of the investment actions of the prototypical agents.

The combination of prototypical agents seems a promising approach to learning virtuous behavior: while individual virtues can be learned using policy regularization, a combination of these virtues might represent a rational

**Fig. 2** Affinity-based RL agent solving a systemic role-playing game. The agent takes virtuous action by optimizing the regularized objective function and receives next state and reward information from the game. Here, the observations 1 to n represents the state. The text highlighted in red represents the affinity of the agent for taking action 2 when encountering a particular combination of observations



agent; we are not equally brave or honorable all the time. This way, an agent actually becomes virtuous rather than utilitarian by being solely dependent on the reward function. The other aspect in virtue ethics is *practical wisdom*, which is to know to what degree an agent must exhibit a virtue depending on the situation. As opposed to the work done in [47], the combinations of virtues may therefore vary in time as well as between individuals. One way of attaining such combinations could be through decision trees with a (partially observable) state vector as input. Another approach could be to extend the policy regularization term in Eq. 1 to specify a state-specific action distribution (Fig. 2), resembling KL-regularization. Formally, the regularization term  $L$  in Equation 1 could be replaced by:

$$L = \sum_{s \in S} \pi_{\theta}(s) \cdot \log \left( \frac{\pi_{\theta}(s)}{\pi_0(s)} \right)$$

Thus, an agent may learn to act honorably in certain states, and bravely in others. Such a prior  $\pi_0$  should specify the desired action distribution as a function of the state variables, e.g., in PP a sick family member might prompt an agent to consider bravery 50% of the time, whereas a dying family member might elicit a higher rate. This is a compelling generalization of global affinity-based RL to local affinity-based RL. Figure 2 illustrates the flow of information from the systemic role-playing game to the policy-regularized deep RL agent. Finally, once the agent is trained to make virtuous decisions in the game, it is crucial to investigate what the agent has learned from these experiences.

#### 4.5 Interpretation of reinforcement learning agents

A virtuous agent is required to perform the right actions for the right reasons; it becomes critical that the decisions

made be scrutinized. At the same time, black-box architectures such as recurrent neural networks (RNN) within the RL framework, are necessary to maintain a good performance. Such a trade-off between interpretability and performance means that an agent must learn to balance between these. In this paper, we use the words “explainability” and “interpretability” interchangeably, but we acknowledge the differences expressed in literature [48]. The composition of prototypical agents is one way of achieving RL interpretability; other methods including causal lens [49], reward decomposition [50] and reward redistribution [51].

The action influence model, introduced by Madumal et al. [49], takes inspiration from cognitive science to encode cause-effect relations by using counterfactuals, i.e., events that could have happened along with the ones that did. We may define the causal model for PP and, based on the action influence model, explain the decisions made by the agent. An alternative approach is the reward decomposition technique, where, in addition to the rewards associated with winning a game, the agent is also incentivised to maximize other reward functions. This maximization is done by decomposing the overall Q-function into multiple elemental Q-functions and calculating differences in rewards using a reward difference explanation technique introduced in [50].

Another interesting approach is the reward redistribution [51], where the expected return is approximated using an LSTM or alignment methods. In reward redistribution, the agent receives delayed rewards at the end of an episode, after every sub-goal, until, finally, the full reward after achieving the main goal. Hence, this approach useful in episodic games such as PP, where salary (reward) is paid at the end of the day, and the main goal of the agent is to keep their family alive using the salary. Finally, apart from the methods mentioned here, we motivate the usage of affinity-based RL for better interpretability since we define the distribution

of virtues in the objective function; it becomes easier to understand the reason for the preference of certain action over others.

## 5 Conclusions and future research directions

In this section, we outline some questions that arise as a result of our work, for instance: how could an artificial agent possibly exhibit virtuous behavior when it clearly lacks human agency and consciousness? At the same time, which virtues are artificial, and which are not? While these questions deserve articles of their own, we attempt to briefly discuss them here. After making the case for virtue ethics, we presented examples of role-playing games such as *Papers, Please* which include ethics as moral dilemmas and we suggested possible approaches to solving such games. Here, we also suggest fruitful directions for future research in virtuous game design and learning algorithms.

We have purposely side-stepped the question of consciousness and moral agency. We are not concerned with conscious artificial agents, but with AI that exists *today*. And once again we stress that the virtues we present here are different from human virtues. For example, in the *Nicomachean ethics* [8], Aristotle argues for the existence of virtues such as temperance and bravery. Such virtues can be thought of exclusively for humans because we show emotions such as anger and fear, whereas at this point, one cannot fathom an artificial agent exhibiting such emotions. Thus, it makes sense to think about a different set of virtues for artificial agents.

Artificial virtues can be thought of as character traits for current day artificial intelligence. A starting point is to consider virtues such as honesty (degree of truthfulness), perseverance (how much to compute), and optimization (how much to fine-tune), demonstrated in [30]. However, unlike [30], we are compelled to progress from mere machine learning towards designing virtuous AI. We consider virtues to be continuous variables; an agent's challenge is to find the golden mean for a given virtue. We will elaborate on this aspect of virtues in a future work.

Previous work has proposed POMDP [16], inverse RL [15] and deep neural network frameworks [7] as possible means to implement artificial virtues. While these are widely adopted models of machine learning, we do recognize that there is a danger that these models might be perceived as consequentialist. There needs to be something more besides the reward function motivating virtuous behaviour. Techniques that work directly on the objective function to encourage certain behaviours may be needed to work in tandem with the reward function. For example, [17] have shown theoretical evidence of agent characterization through policy regularization. Such affinity-based RL methods also aid towards improving the explainability of

models, and this is crucial with respect to virtues, as we highlighted in earlier sections.

Finally, it is important to consider the data or environment used to train such agents, as these influence the model's performance. The framework of systemic role-player games highlighted in *Papers, Please* [13], provides a reasonable model on how to integrate ethical dilemmas into an environment, such that these ethical aspects arise as the agent plays the game and learns to adjust its decision-making based on feedback received from the environment. Depending on the model and the environment used, it may be fruitful to see how multiple virtuous agents behave when they are at odds. Overall, this paper furthers the conversation on the implementation of ethical machines, which is a nascent research area.

## Declarations

**Conflict of interest** The authors have no competing interests.

## References

1. Cervantes, J.-A., López, S., Rodríguez, L.-F., Cervantes, S., Cervantes, F., Ramos, F.: Artificial moral agents: a survey of the current status. *Sci. Eng. Ethics* **26**(2), 501–532 (2020). <https://doi.org/10.1007/s11948-019-00151-x>
2. Formosa, P., Ryan, M.: Making moral machines: why we need artificial moral agents. *AI Soc.* **36**(3), 839–851 (2021). <https://doi.org/10.1007/s00146-020-01089-6>
3. Allen, C., Smit, I., Wallach, W.: Artificial morality: top-down, bottom-up, and hybrid approaches. *Ethics Info. Tech.* **7**(3), 149–155 (2005). <https://doi.org/10.1007/s10676-006-0004-4>
4. Fisher, M., List, C., Slavkovik, M., Winfield, A.: Engineering Moral Agents - from Human Morality to Artificial Morality (Dagstuhl Seminar 16222), 24 (2016). <https://doi.org/10.4230/DAG-REP.6.5.114>
5. Gips, J.: Towards the Ethical Robot. In: Anderson, M., Anderson, S.L. (eds.) *Machine Ethics*, pp. 244–253. Cambridge University Press, Cambridge (2011). <https://doi.org/10.1017/CBO9780511978036.019>. [https://www.cambridge.org/core/product/identifier/CBO9780511978036A028/type/book\\_part](https://www.cambridge.org/core/product/identifier/CBO9780511978036A028/type/book_part)
6. Tolmeijer, S., Kneer, M., Sarasua, C., Christen, M., Bernstein, A.: Implementations in machine ethics: a survey. *ACM Comput. Surveys* **53**(6), 1–38 (2021). <https://doi.org/10.1145/3419633>
7. Stenseke, J.: Artificial virtuous agents: from theory to machine implementation. *AI Soc.* (2021). <https://doi.org/10.1007/s00146-021-01325-7>
8. Ross, W.D., Brown, L. (eds.): *Oxford World's Classics: Aristotle: The Nicomachean Ethics (Revised Edition)* (1980). <https://doi.org/10.1093/actrade/9780199213610.book.1>
9. Red, C.P.: *The Witcher 3: Wild Hunt - Official Website* (2015). <https://www.thewitcher.com/en> Accessed 11 Apr 2022
10. Bethesda Game Studios: *Fallout 4* (2015). <https://fallout.bethesda.net/en/games/fallout-4> Accessed 12 Apr 2022
11. Telltale Games: *Batman: The Telltale Series* (2016). <https://telltale.com/> Accessed 11 Apr 2022
12. Lucas Pope: *Papers, Please* (2013). <https://papersplea.se/> Accessed 11 Apr 2022

13. Formosa, P., Ryan, M., Staines, D.: Papers, please and the systemic approach to engaging ethical expertise in videogames. *Ethics Info. Tech.* **18**(3), 211–225 (2016). <https://doi.org/10.1007/s10676-016-9407-z>
14. Tancred, N., Vickery, N., Wyeth, P., Turkay, S.: Player Choices, Game Endings and the Design of Moral Dilemmas in Games. In: *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pp. 627–636. ACM, Melbourne VIC Australia (2018). <https://doi.org/10.1145/3270316.3271525>. <https://dl.acm.org/doi/10.1145/3270316.3271525>
15. Berberich, N., Diepold, K.: The Virtuous Machine - Old Ethics for New Technology? (2018). <http://arxiv.org/abs/1806.10322>
16. Abel, D., MacGlashan, J., Littman, M.L.: Reinforcement Learning As a Framework for Ethical Decision Making. In: *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, p. 8. AAAI Publications, Phoenix, Arizona, USA (2016). <https://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/view/12582>
17. Maree, C., Omlin, C.: Reinforcement learning your way: agent characterization through policy regularization. *AI* **3**(2), 250–259 (2022)
18. Annas, J.: *Virtue Ethics* (2007). <https://doi.org/10.1093/oxfordhb/9780195325911.003.0019>
19. Alexander, L., Moore, M.: *Deontological Ethics*. The Stanford Encyclopedia of Philosophy (2021). Accessed 5 Apr 2022
20. Sinnott-Armstrong, W.: *Consequentialism* (2003). Last Modified: 2019-06-03. Accessed 17 Nov 2022
21. Hew, P.C.: Artificial moral agents are infeasible with foreseeable technologies. *Ethics Info. Tech.* **16**(3), 197–206 (2014). <https://doi.org/10.1007/s10676-014-9345-6>
22. Foot, P.: *Virtues and Vices* (2002). <https://doi.org/10.1093/0199252866.001.0001>
23. MacIntyre, A.C.: *After Virtue: A Study in Moral Theory* (2007)
24. Moor, J.H.: The nature, importance, and difficulty of machine ethics. *IEEE Intell. Syst.* **21**(4), 18–21 (2006). <https://doi.org/10.1109/MIS.2006.80>
25. Anscombe, G.E.M.: *Modern Moral Philosophy*, 20 (2022)
26. Hursthouse, R.: *On Virtue Ethics* (2001). <https://doi.org/10.1093/0199247994.001.0001>
27. Rosenblueth, A., Wiener, N., Bigelow, J.: Behavior, purpose and teleology. *Philos. Sci.* **10**(1), 18–24 (1943)
28. Wiener, N.: Some moral and technical consequences of automation. *Science* **131**(3410), 1355–1358 (1960). <https://doi.org/10.1126/science.131.3410.1355>
29. Dreyfus, H.L.: *What computers still can't do: a critique of artificial reason*. MIT Press, Cambridge, MA, USA (1992)
30. Coleman, K.G.: Android arete: toward a virtue ethic for computational agents. *Springer* **3**(4), 247–265 (2001)
31. Stenseke, J.: Artificial virtuous agents in a multi-agent tragedy of the commons. *AI Soc.* (2022). <https://doi.org/10.1007/s00146-022-01569-x>
32. Dotnod Entertainment: *Life is Strange* (2022). <https://lifeisstrange.square-enix-games.com/en-gb/> Accessed 11 Apr 2022
33. Nay, J.L., Zagal, J.P.: Meaning Without Consequence: Virtue Ethics and Inconsequential Choices in Games. In: *Proceedings of the 12th International Conference on the Foundations of Digital Games*, pp. 1–8. ACM, Hyannis Massachusetts (2017). <https://doi.org/10.1145/3102071.3102073>. <https://dl.acm.org/doi/10.1145/3102071.3102073>
34. *Origin Systems: Ultima IV: Quest of the Avatar*. *Origin Systems* (1985). [https://en.wikipedia.org/wiki/Ultima\\_IV:\\_Quest\\_of\\_the\\_Avatar#Development](https://en.wikipedia.org/wiki/Ultima_IV:_Quest_of_the_Avatar#Development) Accessed 11 Aug 2022
35. Zagal, J.P.: *Ethically Notable Videogames: Moral Dilemmas and Gameplay*, vol. 5, p. 9. Brunel University, Brunei (2009). <http://www.digra.org/wp-content/uploads/digital-library/09287.13336.pdf>
36. Heron, M.J., Belford, P.H.: Do you feel like a hero yet? Externalized morality in video games. *J. Games Crit.* **1**(2), 25 (2014)
37. Narvaez, D., Vaydich, J.L.: Moral development and behaviour under the spotlight of the neurobiological sciences. *J. Moral Educ.* **37**(3), 289–312 (2008). <https://doi.org/10.1080/03057240802227478>
38. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous Control with Deep Reinforcement Learning. *arXiv* **1509.02971** (2019)
39. Aubret, A., Matignon, L., Hassas, S.: A survey on intrinsic motivation in reinforcement learning. *arXiv* **1908.06976** (2019)
40. Miryoosefi, S., Brantley, K., Daume III, H., Dudik, M., Schapire, R.E.: Reinforcement Learning with Convex Constraints. In: *Advances in Neural Information Processing Systems*, vol. 32, pp. 1–10 (2019)
41. Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. *J. Mach. Learn. Res.* **18**, 1–51 (2015)
42. Lu, J., Dissanayake, S., Castillo, N., Williams, K.: Safety evaluation of right turns followed by U-Turns as an alternative to direct left turns - conflict analysis. Technical report, CUTR Research Reports (2001)
43. Andres, A., Villar-Rodríguez, E., Ser, J.D.: Collaborative training of heterogeneous reinforcement learning agents in environments with sparse rewards: what and when to share? *arXiv* **2202.12174** (2022)
44. Vieillard, N., Kozuno, T., Scherrer, B., Pietquin, O., Munos, R., Geist, M.: Leverage the average: an analysis of KL regularization in reinforcement learning. In: *Advances in Neural Information Processing Systems (NIPS)*, vol. 33, pp. 12163–12174. Curran Associates, Virtual-only (2020). <https://doi.org/10.48550/ARXIV.2003.14089>
45. Haarnoja, T., Tang, H., Abbeel, P., Levine, S.: Reinforcement learning with deep energy-based policies. In: *Proceedings of the 34th International Conference on Machine Learning*, pp. 1352–1361 (2017)
46. Galashov, A., Jayakumar, S., Hasenclever, L., Tirumala, D., Schwarz, J., Desjardins, G., Czarnecki, W.M., Teh, Y.W., Pascanu, R., Heess, N.: Information Asymmetry in KL-Regularized RL. In: *International Conference on Learning Representations (ICLR)*, New Orleans, Louisiana, United States, pp. 1–25 (2019)
47. Maree, C., Omlin, C.W.: Can interpretable reinforcement learning manage prosperity your way? *AI* **3**(2), 526–537 (2022)
48. Heuillet, A., Couthouis, F., Díaz-Rodríguez, N.: Explainability in deep reinforcement learning. *Knowledge-Based Syst.* **214**, 106685 (2021). <https://doi.org/10.1016/j.knsys.2020.106685>
49. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. *Proc. AAAI Conf. Artif. Intell.* **34**(03), 2493–2500 (2020). <https://doi.org/10.1609/aaai.v34i03.5631>
50. Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., Doshi-Velez, F.: Explainable Reinforcement Learning via Reward Decomposition. In: *Proceedings at the International Joint Conference on Artificial Intelligence. A Workshop on Explainable Artificial Intelligence* (2019)
51. Patil, V.P., Hofmarcher, M., Dinu, M.-C., Dorfer, M., Blies, P.M., Brandstetter, J., Arjona-Medina, J.A., Hochreiter, S.: Align-RUD-DE: Learning From Few Demonstrations by Reward Redistribution. *arXiv* (2022). <http://arxiv.org/abs/2009.14108>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.