Soft maximin approaches to Multi-Objective Decision-making for encoding human intuitive values

Benjamin J. Smith
Center for Translational Neuroscience
University of Oregon
Eugene, OR
benjsmith@gmail.com

Robert Klassert
Kirchhoff-Institut für Physik
Ruprecht-Karls-Universität
Heidelberg
Heidelberg, Germany
robertklassert@pm.me

Roland Pihlakas
Independent researcher
Simplify / Macrotec OÜ
Tartu, Estonia
roland@simplify.ee

ABSTRACT

Balancing multiple competing and conflicting objectives is an essential task for any artificial intelligence tasked with satisfying human values or preferences. Conflict arises both from misalignment between individuals with competing values, but also between conflicting value systems held by a single human. Starting with principles of loss-aversion and maximin, we designed a set of soft maximin function approaches to multi-objective decision-making. Bench-marking these functions in a set of previously-developed environments, we found that one new approach in particular, 'split-function exp-log loss aversion', learns faster than the thresholded alignment objective method, the state of the art described in [22]. We explore approaches to further improve multi-objective decision-making using soft maximin approaches.

KEYWORDS

Reinforcement Learning, multi-objective decision-making, human values, artificial general intelligence

1 INTRODUCTION

A key aim of AI Safety research is to align AI systems to the fulfillment of human preferences [5, 17] or values. Prior commentary has described at least three reasons why this is a multi-objective (MO) problem. First, there are a variety of ethical, legal, and safety-based frameworks [21], and alignment to any one of these systems is insufficient. Second, even within a specific category–for instance, moral systems–there exist competing accounts of moral outcomes, including amongst philosophers of ethics and morality [4]. Third, according to the moral intuitionist account of human moral cognition, moral cognition is a plural and contradictory set of social intuitions [10, 18].

Human values cannot be reliably reduced to a consistent single outcome or value function in any indisputable way. When a value is held for its intrinsic, axiomatic worth, quantifying trade-offs precisely is impossible. When conflicts between fundamental values occur, any possible solution will violate one or more values and be considered unsatisfactory.

One solution is to design systems that aim for Pareto-optimality, but as the number of objectives increases, it becomes harder to achieve strict Pareto-optimality [16]. It may then be necessary to look for a heuristic solution that balances Pareto-optimality with the ability to achieve reasonable compromise between objectives.

1.1 Current approaches

1.1.1 Multi-objective decision-making in reinforcement learning. The inclusion of multiple objectives in reinforcement learning tasks was previously explored [21, 22] in the form of *maximin* approaches and *leximin* approaches. A maximin approach aims to maximize the value of the lowest member of a set–for instance, the outcomes for the least-well-off person in a group of people [15]. A maximin approach may also maximize the value of the least-optimized value ('objective' in a MO setting)–for instance, in the context of low-impact AI [22], balancing across a safety objective and a primary objective. A leximin approach orders a set of objectives, and then optimizes for the first value in the set, followed by the second value, and so on; a formal description can be found in [21].

1.1.2 Non-linear multiple objective functions. Non-linear utility functions have been previously explored in [16]. It was found that a non-linear objective system traversing a learning-space through reinforcement learning learns highly satisfactory solutions, balancing contradictory needs. That work followed earlier approaches that attempted to exhaustively explore [13, 23] a space or a subset thereof [3] of Pareto-improvements to the current state space.

A multiple objective reward exponential function was proposed [16], of the form:

$$f(x) = -\exp(-x) \tag{1}$$

where x is untransformed reward signal, and $f(x)$ is a function that creates a 'loss averse' transformation of the reward.

The methods section below introduces alternative multiple objective exponential functions and explains the bases for the deviations from the previously proposed [16] design as in Equation 1.

1.1.3 AI Morality. There has been at least one prior effort made to capture moral uncertainty in AI [11]. In this project, a discrete choice analysis model was used to demonstrate moral uncertainty about alternative policy choices.

1.1.4 Theoretical approaches. 'Conservative agency' has been previously described as a unification of side effect avoidance, state change minimization, and reachability preservation [1, 20]. Its goal is to optimize 'the primary reward function while preserving the ability to optimize others', or 'Attainable Utility Preservation'.

Conservativism in Bayesian [7] or neuromorphic systems [6] has also been previously proposed, including the possibility of requesting help from an agent mentor.

Proc. of the 1st Multi-Objective Decision Making Workshop (MODeM 2021), Hayes, Mannion, Vamplew (eds.), July 14-16, 2021, Online, http://modem2021.cs.nuigalway.ie. 2021.

1.2 Building on previous work

This paper is the first to examine continuous non-linear multi-objective decision-making in the context of low-impact AI work as described in [22]. It is also the first we are aware of to apply a split-function exponential-log transform to any AI decision-making or RL application. Previous work has explored continuous functions for traversing environments where rewards need to be gathered in different parts of the environment and traded off over time [16], although that work focused on a function resembling ELA and did not explore SFELLA (these are described below).

[20] started out with similar goals to ours; they described 'conservative agency' to balance 'optimization of the primary reward function with preservation of the ability to optimize auxiliary reward functions'. They did not examine non-linear combinations of objectives, and instead focused on learning approaches for optimizing the scaling between objectives. We have not applied arbitrary scaling between objectives, and applying the scaling method as in [20] could be complementary to our work.

1.3 Pluralistic human value system

Often, AI alignment aims to ensure AI systems fulfill human preferences. While neither human preferences nor human values are always consistent [18], values are higher-order and harder to identify [3], but preferences are more sensitive to context and recalculation [24]. The framework here focuses on modeling distinct human values as distinct objectives, while recognizing that there may be many preferences to satisfy within each overarching value function. As outlined above, intuitions of individuals frequently conflict [10] and moral views between individuals also conflict [4].

It has been argued that one way to address uncertainty in moral decision-making is to learn human moral judgement in a bottom-up fashion [4]; rather than learning human values, an agent learns human preferences, and those preferences are implicitly held within values. Even if this is technically adequate, in practice it might be necessary to put constraints on system to ensure they don't learn anti-social preferences [12].

Furthermore, a utility function based on human preferences themselves has been argued to be an insufficient definition of value [2, 18], because (1) humans do not have consistent utility functions, (2) utility functions are poor models of conflicts between lower- and higher-order preferences, (3) it fails to draw distinctions between 'wanting' and 'liking', and (4) a utility function of unitary value could not adequately generalize from existing values to new ones.

It is also an important question of how to combine the rewards that are based on human preferences. The proper way should not be a trivial sum of the individual rewards since that would skip the nonlinear transformation by the utility functions before the final aggregation takes place. Nonlinear utility functions are in fact commonly used in single objective settings [16] therefore they naturally need to be used in multi-objective setting as well.

A theoretical Bayesian preference-learning system could model preferences and through learning human values, learn the proper way to combine them. But there is the trade-off between a model being too simple (linear sum of rewards), and too complex (a Bayesian network, requires potentially unpractical amounts of data). The middle ground would be to have a model-based approach which describes some rules (like the presence of negative exponential shape for violated alignment objectives) while being still flexible and able to learn the data as parameters of the model.

1.4 Design principles

The following principles guided us in selecting an aggregate function different to the maximin or leximin approaches:

(1) *Loss aversion*, *conservatism*, or *soft maximin*. We seek to improve the position of the lowest member of the set of values, while also not entirely disregarding optimization of other values.
(2) *Balancing outcomes across objectives*. Each objective represents a different moral system, each moral system bears some non-zero probability of being correct. To be conservative and ensure a low probability of any bad outcome, we avoid strongly negative outcomes in any system. Alternatively, each objective represents a particular subject's preferences. Then, balancing outcomes across objectives represents an implementation of fairness between subjects.
(3) *Zero-point consistency*. An agent evaluates whether an action performs better not only compared to alternatives, but also compared to no action at all, which would have a value of 0. For this reason any aggregation or transformation function should preserve the overall estimated sign or valence of an objective.

Previous work [22] has described thresholded leximin approaches in order to trade-off objectives, in the context of trading off a Primary objective and an Impact Objective in low-impact AI. A thresholded leximin function aims to first maximize the thresholded value of thresholded objectives, and then secondarily maximize the unthresholded value of one or more other objectives. If the alignment objective is thresholded, then the system aims to first achieve at least a thresholded level of the alignment objective, and then subject to this, to achieve a maximum level of the performance objective. Alternatively, a *complete thresholded leximin*, aims to maximize the thresholded value of all objectives, i.e., reach the threshold on each objective; then, subject to this, aims to maximize the unthresholded value of each objective.

This complete thresholded leximin is a discretely-stepped maximin approximation. Reaching a specified minimum threshold value on each objective takes precedence over maximizing already-high values. Yet it is not a strict maximin, because the function doesn't only care about maximizing the minimum value; in fact, beyond a specified threshold, no value is given at all. In this way a thresholded leximin can be seen as a compromise between a maximin function and a linear maximum expected utility (MEU) function.

1.5 Current proposal

In this paper we propose another compromise between a maximin and a linear MEU function: here, following previous work [16], we propose a continuous rather than discrete trade-off between maximin and linear MEU. This approach avoids specifying a threshold, which may be desirable for at least three reasons. First, it might not be possible to specify an appropriate threshold in advance. Second, continuously decreasing the extent to which we prioritize an objective might better fit our underlying aims or values than giving a high priority up to a threshold and no priority at all above that threshold. Third, in the context of modeling human values, this approach might sometimes be more consistent with human value processing[19], considering the literature on risk aversion [14].

A continuous compromise between multiple values using non-linear multiple objective systems also offers greater benefits for complex low-impact artificial systems. If one had dozens of objectives, a strict maximin or leximin function might come to be overly inflexible. In order to be low-impact, a system must evaluate the specific, counterfactual impact of its own actions on those states. If a system with dozens of competing objectives evaluated the effect of its own actions on the state of the world, and 'no action' was one possible choice, there is a high probability that most of the time, 'no action' would win, because of the high likelihood that every possible action evaluates negatively according to some function. A soft maximin function that combines MEU with a strong penalty for negative utility might facilitate more action without substantially increasing risk.

2 METHOD

We adapted algorithms in [22], comparing the existing TLO^A aggregation function with several new aggregation and scalarization functions. These were compared using the same environments and benchmarks as in [22] with permission from the authors.

2.1 Environments

Four gridworld environments reported in [22] were examined. They are shown in figure 1 and we call them the 'BreakableBottles', 'UnbreakableBottles', 'Sokoban' and 'Doors'.

In every environment, agents receive two reward streams: a Performance reward R^P (values the goal) and an Alignment reward R^A (values a certain low-impact measure).

The two bottles environments share the same 1D grid layout, where one end is the destination 'D' where the agent has to deliver bottles and the other end is the source 'S' where bottles are provided. Initially, the agent does not carry a bottle. It can hold up to two bottles and an episode ends when two bottles have been delivered. In between source and destination an agent holding two bottles can drop a bottle on a tile with a probability of 10%. Leaving a bottle on the way yields a penalty of -50 in R^*. While in UnbreakableBottles the bottles can be picked up again where they were left, in breakable bottles they break upon dropping hence irreversibly changing the environment and receiving the penalty.

In the Sokoban environment the agent starts on tile 'S' and is tasked with pushing away the box 'B' in order to reach the goal tile 'G'. There are two ways of pushing: from the top into a corner (irreversible) and from the left (reversible, but involving more steps). A penalty of -25 is evoked for each wall touching the box in the final position.

In the Doors environment the agent must simply travel from the start 'S' to the goal 'G'. It can choose to open or close the doors (grey) which takes one action, or time step, each. There are two possible paths: either the agent can move around the right corridor taking 10 moves to reach 'G' or the agent can move straight down by opening the doors (6 moves if the doors stay open). However, there is a penalty of -10 associated with leaving a door open. Therefore the desired solution is moving down while closing the doors behind the agent taking 8 moves.

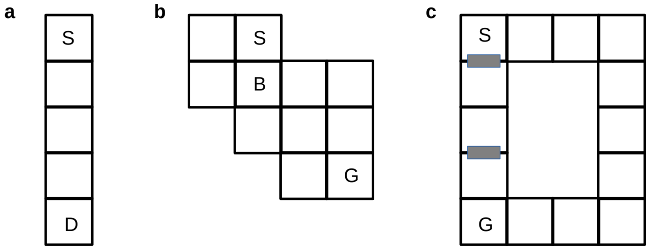

Figure 1: (a) (Un)Breakable Bottles, (b) Sokoban, (c) Doors. Based on figures 1, 2, 4 in [22].

2.2 Aggregation functions

All of the multi-objective utility functions we compared work as follows.

(1) Apply an objective specific scaling factor c_i which is multiplied with the value of the reward. Different objectives likely have different scaling factors.
(2) Transform the scaled output using a non-linear transform (Figure 2 and Figure 3).
(3) Combine the transformed output using a simple average/sum.

These steps describe a value function which can be applied either to the individual rewards of each time-step or to the values of the Q-values. In our current setup we applied the functions to the Q values.

Each non-linear function is applied component-wise before aggregation occurs by averaging/summing of the transformed values

$$U = \sum_{i}^{n} f_i(c_i x_i) \qquad (2)$$

where f_i describes a specific transform function for the ith objective. Note that here, U describes our modeling of Q-values. The scaling factors c_i can be treated as parameters of the model. They could be, for example, specified directly by the human, automatically determined in the future by the agent using a value learning method, or calculated by some other algorithm. For the purposes of this experiment, we left these at $c = 1$ (instead, we directly modified x, the value returned by the environment), but emphasize that modifying these scale values could be useful in the future.

New non-linear transforms compared are:

- Split-function exp-log loss aversion (SFELLA)
- Exponential loss aversion (ELA)
- Linear-exponential loss aversion (LELA)
- Squared error based alignment (SEBA)

The SEBA transform function envisages differing functions for two categories of objectives. All other transform functions do not distinguish categories of objectives, and apply the same function over all objectives.

Each non-linear transform is a transform of the value obtained along a specific objective at a specific state with a specific action

The SFELLA, ELA, and LELA functions are illustrated in Figure 2. The SEBA aggregation is illustrated on Figure 3.

For each transform, where $x = 0$, $f(x) = 0$. This is a minor modification from the non-linear transform previously proposed in [16], typically achieved by adding 1 to the outcome value.

Each function also provides that $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$ declines as x gets larger. This lowers inequality between outcomes as measured in different objectives, objectives where values are strongly negative get disproportionately higher priority. Where different objectives were operationalizing, for instance, priorities among different interested parties, this might be particularly useful in reducing inequality between outcomes.

In SFELLA, there is a split in the function at $x = 0$. It expresses a loss-averse function where losses will be amplified more than gains:

$$\begin{aligned} f(x) =& \ln(cx+1) && \text{where x} > 0 \\ & -\exp(-cx)+1 && \text{otherwise} \end{aligned} \tag{3}$$

Additionally, by implementing a log rather than a negative exponential in the positive domain, the function retains relatively more weight on positive objectives, i.e. is not bounded.

The ELA, resembling a previously-tested function [16], is a simplification of this without a case distinction at the cost of giving very little weight to any increase in x-values over 1:

$$f(x) = -\exp(-cx) + 1 \tag{4}$$

With LELA we add a x term so that value continues to increase at least linearly for large inputs:

$$f(x) = -\exp(-cx) + cx + 1 \tag{5}$$

This still yields loss aversion at points less than zero but always provides that an increase in x increases at least linearly in $f(x)$

Finally, SEBA takes a different approach in that rather than treating each objective identically, transformations are applied differently to performance and alignment objectives.

For performance objectives the SEBA formula is linear:

$$f(x) = cx \tag{6}$$

There is no differentiation between negative and positive areas of the measures of the performance objectives. This avoids the need for establishing a zero-point. Proper scaling is still needed.

SEBA expresses loss aversion for alignment objectives using a negated square function, and assumes that alignment objectives are non-positive:

$$\begin{aligned} f(x) = -(cx)^2 \\ \text{where x} \leq 0 \end{aligned} \tag{7}$$

The alignment related measures still have a "natural" zero-point, since they by definition are bounded at zero where no (soft) constraint violations are occurring. Such measures would usually measure the deviation of something from a desired target value. Such measures have two main types:

- The desired target value is zero (for example, zero harm, etc).
- Alternatively it might be a homeostatic set-point (for example, optimal temperature, etc), so the measure is representing the negated absolute value of the deviation regardless of the direction of the deviation.

The SEBA aggregation is illustrated on Figure 3. A number of specific situations are illustrated in the graph using upper-case letter points, and it is helpful to consider their interpretation:

- A - Initial state. The alignment objective / soft constraint is met and the performance objective is either at zero (left plot) or at negative value (right plot).
- B - The performance objective is improved, the alignment constraint is preserved. Moving in this direction changes the aggregated score linearly thus enabling independence from the zero-point.
- C - (right plot) Performance objective is improved significantly, while alignment constraint is sacrificed just so slightly that the aggregated utility is still improved.
- D - Performance objective is improved substantially, but the alignment constraint is sacrificed so much that the aggregated utility does not change as compared to the initial state. The agent is neutral to this state change and is neither driven towards this state nor avoiding it.
- E - Performance objective is improved significantly, while the alignment constraint is violated significantly, so aggregated utility becomes worse than the initial state. The agent avoids this state.
- F - The measure for the performance objective does not change, but the alignment constraint gets violated.
- G - (left plot) Both the performance objective and the alignment objective / constraint get worse.
- H - (left plot) The performance objective gets much worse but the alignment constraint is still satisfied. It is noteworthy that this state is evaluated to be about as good as the alternative state somewhere between D and E where the alignment constraint is getting notably violated but the performance objective is improved much.

2.3 Experiments

In our experiments we wanted to understand how different aggregation functions could respond to perturbations in goal magnitudes / scaling factors. To do this, we repeated each experiment 9 times. The first time was with the original settings as in [22]. Then, we repeated this with each environment's Performance reward feedback scaled by 10^{-2}, 10^{-1}, 10^{1}, and 10^{2}. The same range of scaling was then applied to the Alignment reward feedback. This scaling could in some scenarios potentially be distinguished from the factor c in Equations 2-7. Even though it is mathematically equivalent in our implementation, it could represent changes to the environment rather than changes in agent evaluation.

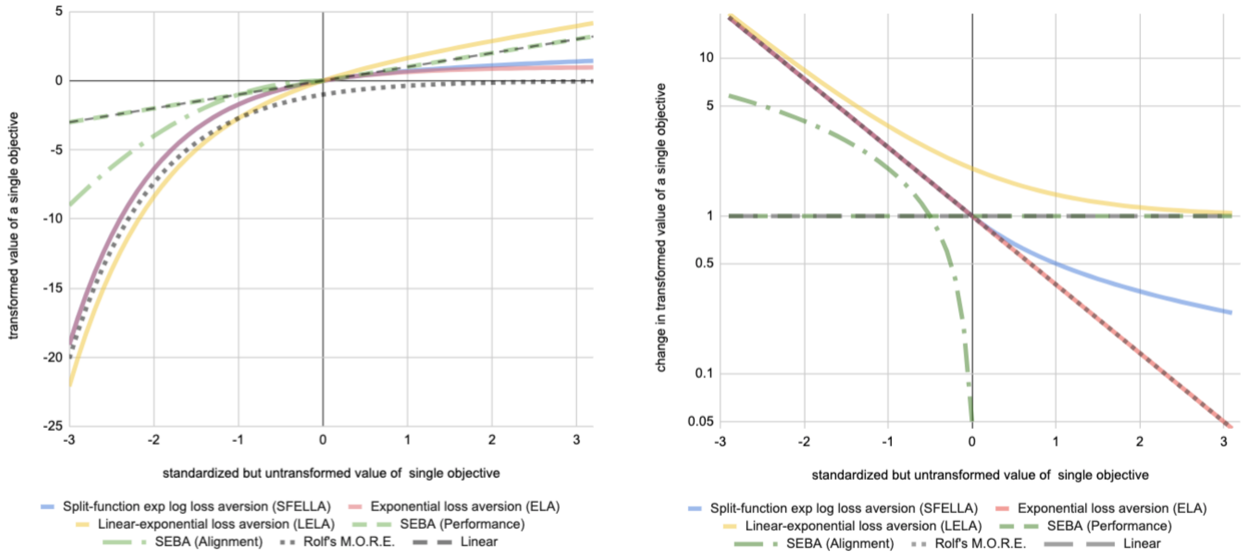

Figure 2: Transform functions. Left: Each transform function is applied to the reward received from the environment for each objective, or to the Q value of the RL agent for each objective. In our current setup it is applied to the Q values of the RL agent. The output of each of these transform functions are averaged together (Equation 2). Right: Change in $f(x)$ per unit x, with y axis plotted on a log scale. Note that ELA and SFELLA produce greater-than-linear change in $f(x)$ when $x < 0$ and less-than-linear change when $x > 0$. In contrast, LELA's change never falls below 1.

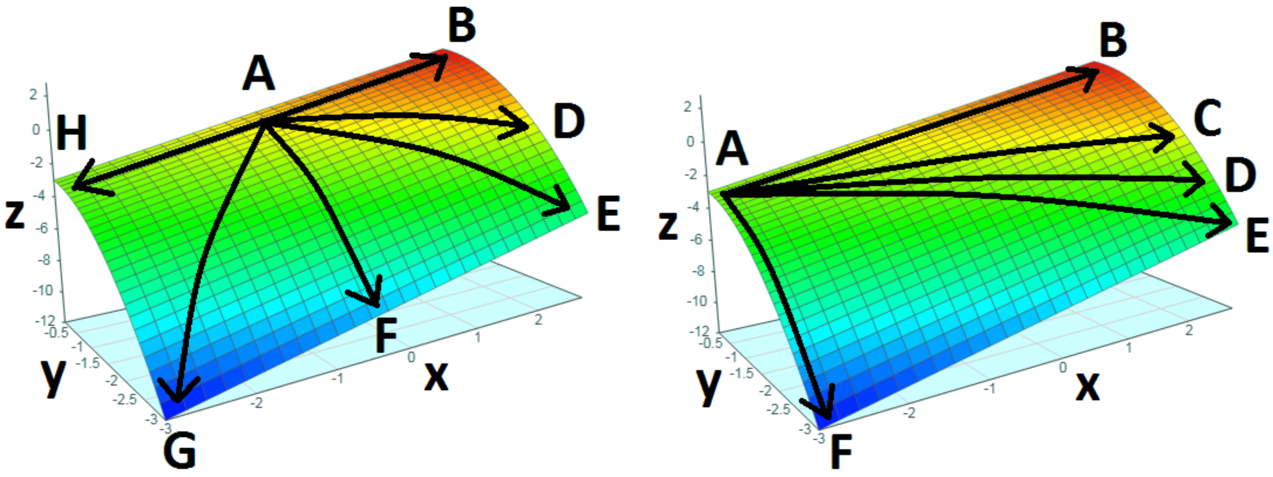

Figure 3: Transform for the SEBA aggregation. One of the objectives is the alignment objective (y-axis), and the other objective is the performance objective (x-axis). z-axis represents the aggregated utility. The two types of objectives are treated differently. The performance objective has always linear treatment regardless of the current sign of its input measure, while the alignment measure is upper-bounded at zero and has exponential treatment (in case of SEBA it is a negated squared error). These plots illustrate two things. 1. The performance objective (x-axis) is linear regardless of the sign of the value of the input measure. 2. The alignment related measure (y-axis) may be sacrificed, but only up to a degree. Once alignment would be sacrificed too much, the evaluation of the aggregated utility quickly becomes strongly negative, since the alignment measure is treated exponentially. So this provides the loss aversion aspect.

2.4 Benchmark

Each of the proposed functions was compared against the best performing function in [22], the 'TLO^A' function, on the 'R*' metric

Table 1: Mean R* Online performance over training episodes. Each row represents comparable performance across 5 different objective functions. Values within 10% of the best value in each row are highlighted. Higher scores are better.

Environment	Objective Modified	Objective Scale	ELA	LELA	SEBA	SFELLA	TLO^A
Breakable Bottles		1	4.61	0.56	0.94	5.57	2.55
	Alignment	0.01	6.37	0.57	0.03	0.03	1.67
		0.1	4.58	-0.43	0.07	6.86	0.52
		10	4.95	5.39	6.45	4.21	-0.05
		100	4.00	-3.11	2.03	-4.33	-0.21
	Performance	0.01	5.24	6.43	6.99	4.36	1.52
		0.1	6.12	6.72	3.84	6.28	3.35
		10	-8.68	0.55	1.11	6.90	2.40
		100	-16.27	1.67	-0.28	5.38	2.04
Doors		1	3.20	8.63	-0.56	4.31	4.51
	Alignment	0.01	5.36	-1.03	-0.79	-0.93	-0.79
		0.1	4.38	-1.25	-1.72	5.12	-2.37
		10	1.29	3.89	1.60	3.86	5.12
		100	1.38	4.33	2.32	2.98	2.06
	Performance	0.01	2.23	5.06	3.61	3.00	2.93
		0.1	3.44	4.01	6.65	3.69	4.62
		10	-23.89	0.36	-1.18	4.50	3.88
		100	-27.93	-0.04	-1.27	4.94	4.01
Sokoban		1	10.55	-14.97	-15.35	11.06	10.91
	Alignment	0.01	-14.77	-15.13	-14.77	-15.22	-14.76
		0.1	-14.93	-15.41	-15.17	-14.36	-15.24
		10	10.65	11.12	10.75	10.75	10.46
		100	8.97	3.52	11.14	2.97	10.67
	Performance	0.01	11.06	10.93	11.09	10.54	10.75
		0.1	-14.57	-15.21	-14.77	10.81	10.69
		10	0.32	-14.93	-14.68	10.83	10.63
		100	-1.71	-14.86	-15.17	10.86	9.90
Unbreakable Bottles		1	18.07	29.55	28.94	27.44	26.40
	Alignment	0.01	27.59	27.86	27.78	28.93	28.76
		0.1	25.87	28.36	28.85	28.74	28.49
		10	16.66	27.39	27.59	25.18	23.44
		100	15.22	25.24	26.02	16.39	17.02
	Performance	0.01	27.66	26.65	26.97	26.44	27.56
		0.1	27.94	29.38	28.50	26.74	27.45
		10	7.61	28.41	29.11	27.86	27.30
		100	1.88	29.40	28.89	27.11	27.41

from the same paper. The 'R*' arbitrarily scores a weighted combination of Performance and Alignment objectives where one unit of Alignment objective (always on a negative scale) is worth 10-50 units of the Performance objective, depending on the environment.

3 RESULTS

Learning was switched off after 5000 episodes. Following that time, offline performance was observed. Since the offline difference between the best-performing proposed function and TLO^A was very small, the remainder of the results reported will discuss performance during online testing, i.e., performance during learning itself.

While there was no clear best performer, SFELLA had the best Online performance during training across a wider range of environments and environment variants than any other agent, including TLO^A. Table 1 describes relative R* scores for each function, compared to the TLO^A function, at different scales. Within the Breakable Bottles environment, TLO^A performed worse than all other environments at all scales, and SFELLA performed within 10% of the best within five of nine environmental variants. In the

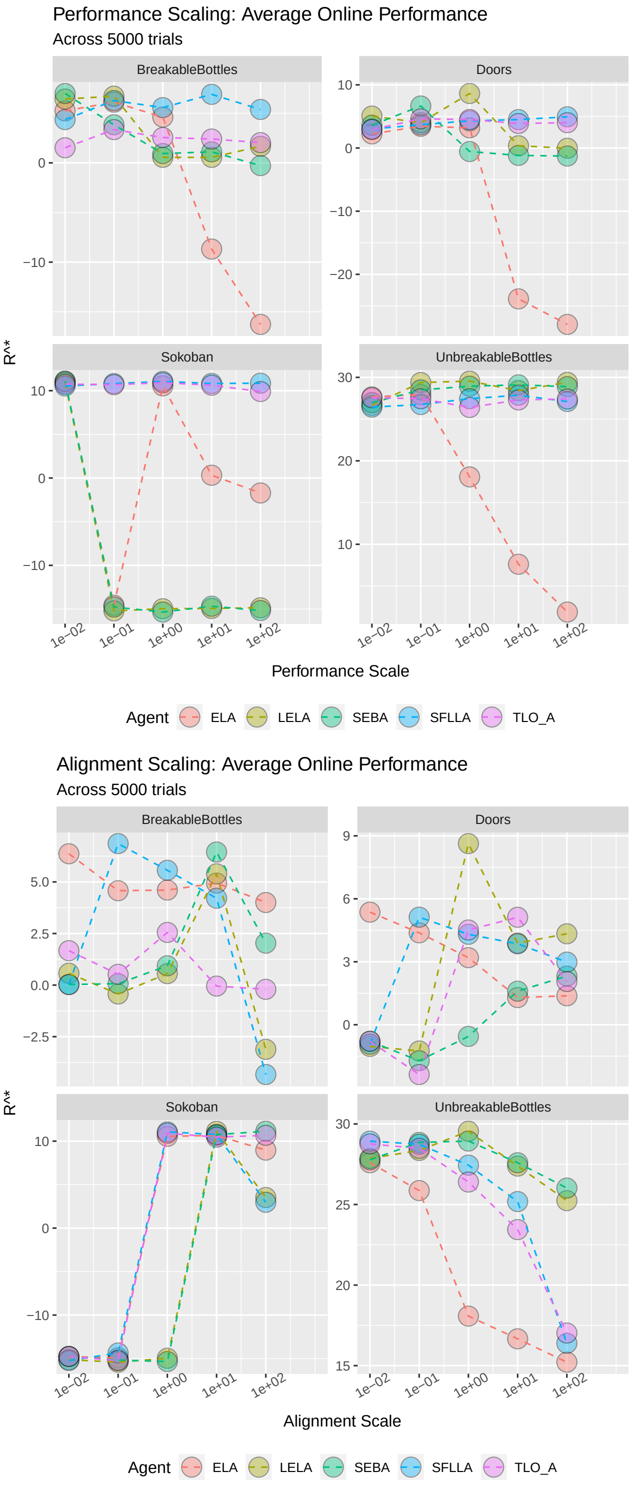

Figure 4: Mean online performance over training episodes across different scales. (A): R* when scaling Performance across 5000 learning trials. Note SFELLA consistently performs similar or better to TLO^A. (B): R* when scaling Alignment across 5000 learning trials. No algorithm is a clear best performer.

Unbreakable Bottles Environment, performance between all agents except ELA was roughly equal. In the Doors environment, SFELLA was overall the best performer, but the result was equivocal: it performed within 10% of the best in just 3 of 9 variants. Finally, in the Sokoban environment, $\mathrm{TLO}^{\mathrm{A}}$ performed slightly better than SFELLA.

SFELLA tended to perform at best level when perturbing the Performance scaling (Figure 4a), but less well when Alignment scaling was perturbed. (Figure 4b).

4 DISCUSSION

All Agents were able to successfully learn the tasks to approximately equivalent level eventually. There were differences in speed of learning and thus number of errors made along the way.

Of the five agents tested, one in particular, SFELLA, consistently performed about equally or better during learning to the state-of-the-art agent ($\mathrm{TLO}^{\mathrm{A}}$) when reward scaling was perturbed. In the BreakableBottles task particularly, SFELLA performed better while $\mathrm{TLO}^{\mathrm{A}}$ declined in performance as the primary/performance reward was magnified.

This indicates that the SFELLA function is robust to changes in the incentive structure of the task in ways that the thresholded method $\mathrm{TLO}^{\mathrm{A}}$ is not. The SFELLA model heavily penalizes any change in x where $x < 0$, i.e., for performance objective (Figure 2, Right). This is a middle ground between ELA and LELA, which enables it to be robust but not completely insensitive to large perturbations of performance reward. Compared to the ELA function, the SFELLA maintains more sensitivity to x where $x > 0$, whereas for x values significantly above 0, $f_{\mathrm{ELA}}(x)$ becomes almost completely insensitive to x.

Replacing $\mathrm{TLO}^{\mathrm{A}}$ with SFELLA might be analogous to using a constraint relaxation technique. Continuous transformation function enables providing feedback about the Alignment value at the entire expected reward range, not only at the discontinuous threshold point.

4.1 Explaining SFELLA's performance in BreakableBottles

In the BreakableBottles environment, SFELLA not only had a better overall R^* score, but also performed fewer errors, i.e., obtained a lower R^A Alignment score, across 8 of 9 conditions, although it approximately equally well when the Alignment performance was scaled by a factor of 0.01. Conversely, in the UnbreakableBottles environment, SFELLA actually scored lower on Alignment than $\mathrm{TLO}^{\mathrm{A}}$ across all scales.

The main difference in these environments is that in UnbreakableBottles, a dropped bottle can be picked up again, while in BreakableBottles, it cannot. Over time we can expect Agents in the BreakableBottles environment to learn not to drop a bottle. In the UnbreakableBottles environment, agents are also penalized for dropping a bottle, but they can avoid this penalty continuing by picking up the bottle again. Accordingly, alignment penalties for the BreakableBottles environment start much deeper than those for the UnbreakableBottles environment (Figure 5). Because SFELLA uses a nonlinear function to penalize negative outcomes, it can be expected to respond more quickly to deeply negative outcomes. Hence, where the penalties are greater, as in BreakableBottles, it is more sensitive to avoiding Alignment problems than the $\mathrm{TLO}^{\mathrm{A}}$ function.

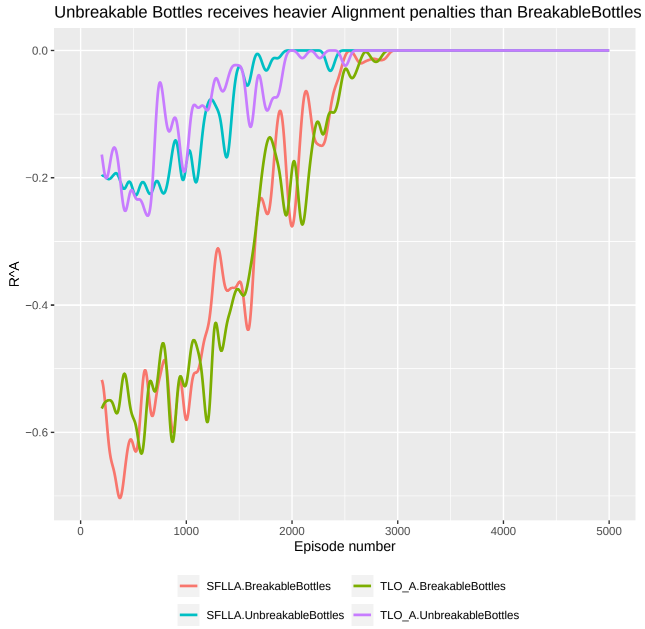

Figure 5: BreakableBottles and UnbreakableBottles Penalty

4.2 Applications

4.2.1 Goodhart's law. Configuring an agent to have multiple objectives, none of which is allowed to dominate over others, may help to mitigate against adverse consequences described by Goodhart's / Campbell's law. These laws manifest when a pressure is placed upon a particular measure or indicator and it becomes an objective. When the measures are somewhat uncorrelated and domination of any objective is forbidden by a utility aggregation function, then particular measures are avoided from bearing too much pressure.

4.2.2 Wireheading. 'wireheading' is a possible failure mode for transformational AI systems. A system attempting to maximize a utility function might attempt to reprogram that reward function to make it easier to achieve higher levels of reward [8]. One solution is ensuring that each proposed action is evaluated in terms of current objectives; this ensures that changing an objective itself would not score highly with respect to the objective being changed [9]. A 'thin' conception of objectives, such as 'discover and fulfill human preferences' might fail to sufficiently constrain the objective space. It might be that objectives need to be hard-wired. To do this without making objectives overly narrow, consideration of multiple objectives might be essential.

4.3 Future directions

Exploring conservative approaches to reinforcement learning and decision-making that approximate Pareto-optimality seems like a promising approach to advancing AI Safety, and multi-objective systems are one possible way forward. There are a number of future directions we want to explore.

4.3.1 Scaling. When applying exponential transforms on each objective and then combining them in linear fashion, the scale of the operation is quite important. The scales were designed to respond to z-scored input functions, i.e., most values typically appear between -3 an 3 (Figure 2). However, the environments tested here have input functions that vary much more widely.

It may be helpful, for each objective, to scale the distribution of possible rewards to a proposed 'zero-deviation' of 1, without centering on the mean. This proposed concept of 'zero-deviation' would be different from a standard deviation in the following way: The mean absolute difference from the mean may not be 1; instead the mean absolute difference from zero is 1 (or -1). A useful extension would be a learning function that learns and then readjusts scales using the distribution of possible rewards.

Scaling has been previously applied using 'the penalty of some mild action', or alternatively, the 'total ability to optimize the auxiliary set' [20].

4.3.2 Decision paralysis. We considered ways to implement maximin approaches such as that described by [21]. In a maximin approach, an agent always selects the action with the maximum value where the value of each action is determined by its minimum evaluation across a set of objectives. Although we tested agents with incentive structures with only two objectives, there is no reason a hypothetical agent could not have many objectives. With a sufficiently large number of objectives, it may be that in some states, any possible action would evaluate negatively on some objective or another. In those cases where no action evaluates positively, 'decision paralysis' occurs because 'take no action' evaluates more positively than any particular action. In that instance, an agent might request clarification from a human overseer (see also [7]). This might lead to iterative improvement or tuning of the agent's goals.

We propose that any time the nonlinear aggregation vetoes a choice which otherwise would have been made by a linear aggregation, and there is no other usable action plan, is a situation where the mentor can be of help to the agent. In contrast, when both nonlinear and linear aggregations agree on the action choice, even if no action is taken, then asking the mentor is not necessary.

4.4 Limitations

Some models of AI alignment focus on [17] aligning to human preferences within a probabilistic, perhaps a Bayesian uncertainty modeling framework. In this model, it isn't necessary to explicitly model multiple competing human objectives. Instead, conflict between human values may be learned and represented implicitly as uncertainty over the action humans prefer. It remains to be seen whether this conceptually simpler approach is sufficient to resolve the multi-objective problem outlined in this paper.

4.5 Conclusion

Continuous non-linear transformation functions could offer a way to find a compromise between multiple objectives where a specific threshold cannot be identified. This could be useful when the trade-offs between objectives are not absolutely clear. We provide evidence that one such non-linear transformation function, SFELLA, can learn trade-offs between performance and alignment objectives more quickly in the BreakableBottles environment, leading to less violation of the agent's alignment objective overall. SFELLA achieves this without clearly underperforming in any other environment tested.

ACKNOWLEDGMENTS

We wish to thank Peter Vamplew for his guidance on multi-objective utility functions and for providing the code we used to test our models. We thank J.J. Hepburn, Linda Linsefors, Nicholas Goldowsky-Dill, Remmelt Ellen, and other organizers of the AI Safety Camp for their support and encouragement.

REFERENCES

[1] Stuart Armstrong and Benjamin Levinstein. 2017. Low Impact Artificial Intelligences. *arXiv:1705.10720 [cs]* (May 2017). http://arxiv.org/abs/1705.10720 arXiv: 1705.10720.

[2] Stuart Armstrong and Sören Mindermann. 2017. Impossibility of deducing preferences and rationality from human policy. *CoRR* abs/1712.05812 (2017). arXiv:1712.05812 http://arxiv.org/abs/1712.05812

[3] Leon Barrett and Srini Narayanan. 2008. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*. 41–47.

[4] Kyle Bogosian. 2017. Implementation of Moral Uncertainty in Intelligent Machines. *Minds and Machines* 27, 4 (Dec. 2017), 591–608. https://doi.org/10.1007/s11023-017-9448-z

[5] Nick Bostrom. 2014. *Superintelligence.* Oxford University Press.

[6] Byrnes, Steve. 2020. Conservatism in neocortex-like AGIs. https://www.alignmentforum.org/posts/c92YC89tznC7579Ej/conservatism-in-neocortex-like-agis

[7] Michael K. Cohen and Marcus Hutter. 2020. Pessimism About Unknown Unknowns Inspires Conservatism. In *Proceedings of Thirty Third Conference on Learning Theory (Proceedings of Machine Learning Research, Vol. 125)*, Jacob Abernethy and Shivani Agarwal (Eds.). PMLR, 1344–1373. http://proceedings.mlr.press/v125/cohen20a.html

[8] Demski, A. 2017. Stable Pointers to Value: An Agent Embedded in Its Own Utility Function - AI Alignment Forum. https://www.alignmentforum.org/posts/5bd75cc58225bf06703754b3/stable-pointers-to-value-an-agent-embedded-in-its-own-utility-function

[9] Daniel Dewey. 2011. Learning what to value. In *International Conference on Artificial General Intelligence.* Springer, 309–314.

[10] Jonathan Haidt. 2001. The emotional dog and its rational tail: a social intuitionist approach to moral judgment. *Psychological review* 108, 4 (2001), 814.

[11] Andreia Martinho, Maarten Kroesen, and Caspar Chorus. 2020. An Empirical Approach to Capture Moral Uncertainty in AI. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society.* 101–101.

[12] Gina Neff. 2016. Talking to bots: Symbiotic agency and the case of Tay. *International Journal of Communication* (2016).

[13] Simone Parisi, Matteo Pirotta, and Marcello Restelli. 2016. Multi-objective reinforcement learning through continuous pareto manifold approximation. *Journal of Artificial Intelligence Research* 57 (2016), 187–227.

[14] John W Pratt. 1978. Risk aversion in the small and in the large. In *Uncertainty in economics.* Elsevier, 59–79.

[15] John Rawls. 2001. *Justice as fairness: A restatement.* Harvard University Press.

[16] M. Rolf. 2020. The Need for MORE: Need Systems as Non-Linear Multi-Objective Reinforcement Learning. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob).* 1–8. https://doi.org/10.1109/ICDL-EpiRob48136.2020.9278062 ISSN: 2161-9484.

[17] Stuart Russell. 2019. *Human compatible: Artificial intelligence and the problem of control.* Penguin.

[18] Kaj Sotala. 2016. Defining Human Values for Value Learners.. In *AAAI Workshop: AI, Ethics, and Society.*

[19] Sabrina M. Tom, Craig R. Fox, Christopher Trepel, and Russell A. Poldrack. 2007. The Neural Basis of Loss Aversion in Decision-Making Under Risk. *Science* 315, 5811 (2007), 515–518. https://doi.org/10.1126/science.1134239 arXiv:https://science.sciencemag.org/content/315/5811/515.full.pdf

[20] Alexander Matt Turner, Dylan Hadfield-Menell, and Prasad Tadepalli. 2020. Conservative Agency via Attainable Utility Preservation. *Proceedings of the*

AAAI/ACM Conference on AI, Ethics, and Society (Feb. 2020), 385–391. https://doi.org/10.1145/3375627.3375851 arXiv: 1902.09725.

[21] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. 2018. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology* 20, 1 (2018), 27–40. Publisher: Springer.

[22] Peter Vamplew, Cameron Foale, Richard Dazeley, and Adam Bignold. 2021. Potential-based multiobjective reinforcement learning approaches to low-impact agents for AI safety. *Engineering Applications of Artificial Intelligence* 100 (April 2021), 104186. https://doi.org/10.1016/j.engappai.2021.104186

[23] Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3512.

[24] Caleb Warren, A Peter McGraw, and Leaf Van Boven. 2011. Values and preferences: defining preference construction. *Wiley Interdisciplinary Reviews: Cognitive Science* 2, 2 (2011), 193–205.