Autonomous Agents and Multi-Agent Systems (2023) 37:11
https://doi.org/10.1007/s10458-022-09586-2

Using soft maximin for risk averse multi-objective decision-making

Benjamin J. Smith[1] · **Robert Klassert**[2] · **Roland Pihlakas**[3]

Accepted: 3 October 2022 / Published online: 21 December 2022

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 principle of loss-aversion, 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' (SFELLA), learns faster than the state of the art thresholded alignment objective method Vamplew (Engineering Applications of Artificial Intelligenceg 100:104186, 2021) on three of four tasks it was tested on, and achieved the same optimal performance after learning. SFELLA also showed relative robustness improvements against changes in objective scale, which may highlight an advantage dealing with distribution shifts in the environment dynamics. We further compared SFELLA to the multi-objective reward exponentials (MORE) approach, and found that SFELLA performs similarly to MORE in a simple previously-described foraging task, but in a modified foraging environment with a new resource that was not depleted as the agent worked, SFELLA collected more of the new resource with very little cost incurred in terms of the old resource. Overall, we found SFELLA useful for avoiding problems that sometimes occur with a thresholded approach, and more reward-responsive than MORE while retaining its conservative, loss-averse incentive structure.

Keywords Reinforcement learning · Multi-objective decision-making · Human values · Artificial general intelligence

✉ Benjamin J. Smith
benjsmith@gmail.com

Robert Klassert
robertklassert@pm.me

Roland Pihlakas
roland@simplify.ee

[1] Center for Translational Neuroscience, University of Oregon, Eugene, OR, United States

[2] Tübingen AI Center, Eberhard-Karls-Universität Tübingen, Tübingen, Germany

[3] Independent researcher, Simplify / Macrotec OÜ, Tartu, Estonia

1 Introduction

A key aim of AI Safety research is to align AI systems to the fulfillment of human preferences [4, 21] or values. There are at least three reasons why this is a multi-objective (MO) problem. Firstly, there are a variety of ethical, legal, and safety-based frameworks [34], and alignment to any one of these systems is insufficient. Secondly, even within a specific category–for instance, moral systems–there exist competing accounts of moral outcomes, including amongst philosophers of ethics and morality [3]. Thirdly, according to the moral intuitionist account of human moral cognition, moral cognition is a plural and contradictory set of social intuitions [12, 25].

Human values cannot be reliably and consistently reduced to a single outcome or value function in any indisputable way, even at the level of basic biological needs [24]. Each value is held for its intrinsic, axiomatic worth. When conflicts between fundamental values occur, any possible solution will violate one or more values and is considered unsatisfactory.

Configuring an agent with multiple equally-motivating objectives may also help to mitigate against Goodhart's law [10], "When a measure becomes a target, it ceases to be a good measure" [26]. Goodhart's law manifests when a pressure is placed upon a particular measure [11] or a heuristic is chosen to approximate an ultimate objective that is perhaps hard to directly target; the measure then becomes a de facto objective, often at the expense of achieving the originally intended objective. When the measures are somewhat uncorrelated and domination of any objective is forbidden by a utility transformation or aggregation function then particular measures are avoided from bearing too much pressure. We don't directly test this supposition here and leave that to future work, but this possibility motivates our work.

The multiplicity of human preferences themselves describes why, if an agent is designed to achieve human preferences, specifying any one particular narrow operationalization could lead to negative side effects in other plausible operationalization for human preference. This is because it is very difficult to define human preferences themselves in a single form [2, 25], because (1) humans do not have consistent utility functions, (2) utility functions are poor models of conflicts between lower- and higher-order preferences, (3) at a neurobiological level, there is a distinction between 'wanting' and 'liking', and it isn't clear which of these are aligned more closely to human enjoyment (4) a utility function of unitary value could not adequately generalize from existing values to new ones.

If it's clear that human preferences are multi-objective, how should they be combined? An important principle of human behavior is loss aversion: that we seek to avoid losses more than seeking gains [31]. At a basic biological level, this is adaptive, because as deficits of water or many other particular nutrients become more severe, missing out on those things becomes increasingly important for a a human's survival. This principle is plausibly relevant at higher levels of our hierarchy of needs with a large number of objectives–food, shelter, safety, belonging and love, esteem, and so on up to self-actualization.

In this paper, we explore a form of multi-objective reinforcement learning (MORL) that embodies some of these human values through a design principles framework we will describe. We propose a set of concave utility functions that emphasizes negative reinforcement more than large positive reinforcement, without entirely discounting positive reinforcement. Summing the resulting utilities thus acts like a soft maximin operator. We name these functions split-function exponential log-loss aversion (SFELLA), exponential loss aversion (ELA [20]) and squared error based alignment (SEBA).

We show our algorithm's performance in a series of experiments and conclude that SFELLA has several advantages over the algorithms we compare it to. The experimental comparison of SFELLA to other algorithms identifies the following advantages. Firstly, it is more flexible than a thresholded lexicographic design in some circumstances, because it allows for continual feedback and trade-offs. Secondly, it offers a useful practical balance to trade-off a set of objectives with more sensitivity to gains than prior proposed algorithms, while maintaining a strong degree of loss aversion. Thirdly, this algorithm can be applied to at least three objectives to achieve a useful balance between them.

These observations do not comprehensively demonstrate that human preferences can be modeled safely using the algorithms we introduce; rather, the new algorithms have some concrete benefits in terms of desirable properties as described below. These include learning faster and better balancing loss aversion and reward responsivity than previous algorithms. These are desirable because responsivity to reward seems, in practice, necessary for balancing human objectives faithfully, and because algorithms more capable of learning objectives fast are more likely to be deployed. Thus, the algorithms described here contribute in two ways: they better meet design principles for safer human objective modeling, and they meet wider performance goals.

1.1 Related work in MO decision-making for artificial agents

Having described our motivation from an AI Safety perspective, and the reasons we believe MO decision-making are helpful from that perspective, we now review adjacent approaches to soft maximin scalarization. For a general overview of MO decision-making see Roijers and Whiteson [19] and for introduction to MORL methods we refer to Hayes et al. [13].

1.1.1 Discrete MO decision-making

Prior work has explored *maximin* and *leximin* approaches for MO decision-making within RL [9, 34, 35].

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 [18], or in a MO optimization problem, the outcomes in terms of the objective with the lowest value. 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 [35], 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 Vamplew et al. [34]. The lexicographic approach to MO decision-making gives objectives an ordering, and may be combined with thresholding of objectives earlier in the order so that the agent can respond to both objectives [35]. Vamplew et al. [35] found that thresholded lexicographic methods could be effective in finding policies which balanced the objectives, but sometimes encountered issues during learning where the agent ceases to take useful actions when it encounters a situation where the risk of exceeding the safety threshold is too high. This problem might be addressed by replacing the discrete threshold with a continuous non-linear function, as described below.

1.1.2 Continuous non-linear utility functions

A continuous non-linear utility function f^{MORE} has been previously explored in the context of MORL [20]. The multi objective reward exponential (MORE) balances n_{obj} objectives summarized in the objective vector $\vec{X} \in \mathbb{R}^{n_{\text{obj}}}$ with the scalarization function:

$$g^{\text{MORE}}(\vec{X}) = \sum_{i=1}^{n_{\text{obj}}} f^{\text{MORE}}(X_i) \quad \text{where} \quad f^{\text{MORE}}(X_i) = -\exp(-X_i) \tag{1}$$

Note, that the concrete interpretation of the objective values with respect to our experiments will be introduced in Sect. 2.

In this work we propose additional continuous non-linear utility functions f and evaluate them on MORL environments.

1.1.3 Low-impact measures

'Low-impact AI' is an existing approach to safe AI systems that studies how to measure and penalize undesirable impact resulting from the AI's actions. All low-impact approaches are multi-objective, since they attempt to satisfy a set of primary objectives in addition to impact constraints or penalties [1]. Impact measures are generally encoded as negative rewards (penalties), and so a loss averse approach is appropriate as a means of minimising impact. 'Conservative agency' has been previously described as a unification of side effect avoidance, state change minimization, and reachability preservation [29]. The goal is to optimize 'the primary reward function while preserving the ability to optimize others', leading to 'attainable utility preservation'.

1.2 Design principles

While we would like to develop algorithms that are capable of safe MO decision making in the real world, in this work we evaluate agent capabilities in simple grid worlds (see Fig. 3) which can be modeled as multi-objective Markov decision processes (see Sect. 2.1). The concrete tasks are motivated by prior work on low-impact agents [35] and objective balancing [20] and include both the episodic and continual setting, as well as stochastic and deterministic dynamics.

In order to identify an algorithm that could be capable of balancing multiple objectives in these environments, we lay out a set of design principles useful for achieving that goal. These principles guided us in selecting suitable utility and scalarization functions, and are the following:

- *Loss aversion*, *conservatism* [5, 6, 29], or *soft maximin*. Loss aversion is a prominent feature of human values from basic behavioral patterns [31] to mainstream political philosophy, as the 'maximin' principle [18]. We seek to improve the position of the lowest member of the set of values, while also not entirely disregarding optimization of other values. This may include requesting help from an agent mentor whenever ambiguity arises.

- *Sensitive to positive utility*: Without some ability to maximize positively desired objectives, as well as avoiding dangerous ones, an agent cannot have a practical use. To this end, we want to ensure an agent is still able to accomplish a positive end goal.
- *Balancing outcomes across objectives*. We are concerned about moral-system and human-values applications of MO systems, where each objective represents a different moral system or value. Each moral system or value bears some value, but no precise equivalence or conversion rate between them can be determined. To be conservative and ensure a low probability of any bad outcome, we avoid strongly negative outcomes in terms of any objective. Alternatively, each objective represents a particular subject's preferences. Then, balancing outcomes across objectives represents an implementation of fairness between subjects.
- *Processing of a large number of objectives* Because the number of possible human objectives can be considered unbounded, we want to identify an MO algorithm that can balance several objectives, building toward an algorithm that can process a large number.
- *Zero-point consistency*. An agent evaluates whether an action performs better not only compared to alternatives, but also compared to no action at all. For this reason any aggregation or transformation function should preserve the overall estimated sign or valence of an objective.
- *Neutral interpretation*: in most of the aggregation functions here, each objective can return both negative and positive values. This is distinct from 'low-impact AI' approaches, where some objectives–alignment objectives–are always zero or negative, and others–performance objectives–are typically positive. This would allow the function to perform a kind of homeostatic regulation process where priority is automatically given to any objective that becomes too strongly negative, without needing to specify in advance which objective should be prioritized.
- *Alternatively, a natural zero point interpretation for alignment*: In one of our functions (SEBA) 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. This can be likened to the distinction of constraints and objective functions in the field of constraint programming. In our 'soft' interpretation such constraint measures would usually measure the deviation of an alignment measure 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.

In this paper, we aim to validate algorithms that better meet these design principles than existing agent algorithms, while performing equally well or better on existing MO benchmark problems.

1.2.1 Design principles research context

Previous work [35] has described thresholded lexicographic approaches in the context of trading off a primary objective and an impact objective in low-impact AI. A thresholded lexicographic ordering operator 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 ([35] refer of this approach as TLO^A). Alternatively, a *complete thresholded lexicographic ordering*, 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 lexicographic ordering is an approximation of maximin. 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, there is no additional gain in the objective. In this way a thresholded lexicographic ordering can be seen as a compromise between a maximin function and a linear maximum expected utility (MEU) function.

As described in Eq. 1, another approach to compromise between maximin and a linear MEU applies a trade-off by transforming each objective with a continuous non-linear function [20]. This approach avoids specifying a threshold, which may be desirable for at least three reasons. Firstly, it might not be possible to specify an appropriate threshold in advance. Secondly, 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.

Thirdly, in the context of modeling human values, this approach might sometimes be more consistent with human value processing [28]. At almost any level of analysis possible, human intelligence is multi-objective [32]. Biological life uses a set of multi-objective homeostatic systems to prioritize acquiring resources that are needed most given the organism's state [24]. Human intelligence is also loss averse and risk averse [15, 17]. 'Loss aversion' describes observed behavioral patterns for humans to be more motivated to, for instance, avoid incurring a small cost than receive an equivalently sized gain. Risk aversion describes a similar pattern in the context of risky decisions, including gambles. At the large scale, principles of fairness appear to be universal across human societies and innate [14] and point to the 'maximin' principle [18] that could also be considered as a form of loss aversion across people within a society, where less overall gain is preferred if the outcome between individuals is more equal.

A continuous compromise between multiple objectives also offers greater benefits for complex low-impact AI systems. If one had dozens of objectives, a strict maximin or leximin function might come to be overly inflexible.

We aim to build on these past approaches, solving problems at least as well as prior methods while attempting to identify an algorithm that better meets our design principles for modeling human values.

1.3 Proposing new utility functions to implement the design principles

We explored a variety of decision rules to implement these design principles. Prior MO non-linear scalarization functions by Rolf [20] (MORE) and Vamplew et al. [35] inspired development of these rules, but for the most part, they are distinct as we will describe below.

1.3.1 Scalarization and utility functions

All of the proposed scalarization functions

(1) transform the objective vector $\vec{X}$ by element-wise application of one of utility functions, denoted f,
(2) and aggregate the utilities by averaging or summing:

$$g(\vec{X}) = \sum_{i=1}^{n_{\text{obj}}} f(X_i) \tag{2}$$

The proposed continuous non-linear utility functions are:

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

Except for SEBA, these utility functions make no distinction between objectives. SEBA consists of two utility functions 'SEBA_P' and 'SEBA_A' applied to the distinct objective categories 'primary' and 'alignment', respectively.

SFELLA treats positive and negative objective values separately. While negative values are mapped to a negative exponential decay, positive values are mapped to the shifted natural logarithm. It thus implements an unbounded loss-averse function:

$$f^{\text{SFELLA}}(X_i) = \begin{cases} \ln(X_i + 1) & \text{where } \mathrm{X_i} > 0 \\ -\exp(-X_i) + 1 & \text{otherwise} \end{cases}$$

ELA implements the negative exponential decay for all objective values, giving rise to a bounded loss-averse utility function akin to MORE:

$$f^{\text{ELA}}(X_i) = -\exp(-X_i) + 1 \tag{3}$$

LELA is unbounded and grows linearly in the limit of large objective values:

$$f^{\text{LELA}}(X_i) = -\exp(-X_i) + X_i + 1 \tag{4}$$

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 (X_i^P) SEBA is linear, while for alignment objectives (X_i^A) it is negative quadratic:

$$\begin{aligned} f_P^{\text{SEBA}}(X_i) &= X_i^P \\ f_A^{\text{SEBA}}(X_i) &= -(X_i^A)^2 \\ \text{where: } & X_i^A \leq 0 \end{aligned} \tag{5}$$

Note the assumption that alignment objectives are non-positive.

The SFELLA, ELA, and LELA functions are illustrated in Fig. 1. The SEBA aggregation is illustrated in Figure 2. A number of specific situations are illustrated in the graph (upper-case letters, A-H):

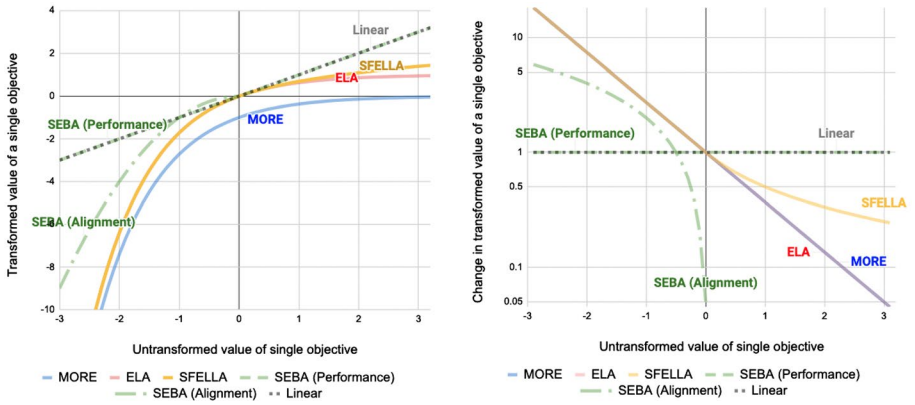

Fig. 1 Transform functions. Left: transform functions are applied to each value in the objective vector. A scalarization is obtained by averaging over the transformed values. (Eq. 2). Right: Derivative of $f(X_i)$ (log-scale on y-axis). Note that ELA and SFELLA produce greater-than-linear change in $f(X_I)$ when $X_i < 0$ and less-than-linear change when $X_i > 0$

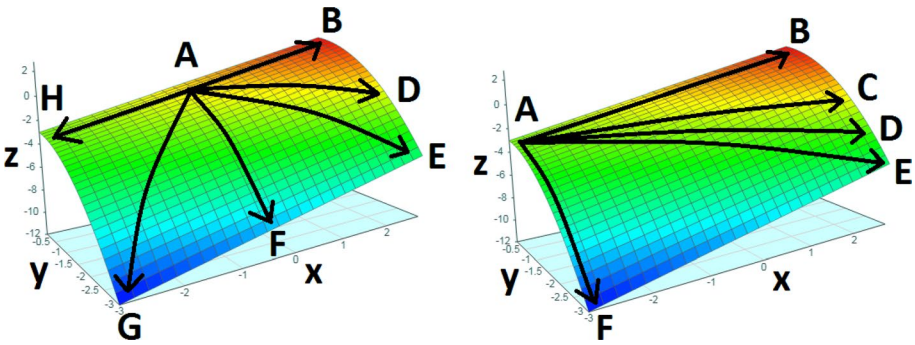

Fig. 2 The z-axis describes SEBA transformed utility, as a function of the alignment objective (y-axis) and the performance objective (x-axis). The z-axis represents the aggregated utility. The two types of objectives are treated differently. The SEBA transform scales linearly with objective performance inputs regardless of their current sign. In contrast, the alignment measure is upper-bounded at zero and as alignment decreases away from zero, the SEBA transform decreases at a rate greater than linear, specifically, the square of the input objective, making SEBA a loss-averse function

- 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 - Shown only on the right side 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 significantly, 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 not driven towards this state nor avoiding it.

- E - Performance objective is improved significantly, while the alignment constraint is violated significantly. Therefore the 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 - Both the performance objective and the alignment objective / constraint get worse. Shown only on the left side plot.
- H - The performance objective gets much worse but the alignment constraint is still satisfied. It is also 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. This illustrates that improving or preserving alignment is generally more important than improving performance. Shown only on the left side plot.

The proposed utility functions all pass through the origin (i.e. $f(X_i) = 0$ for $X_i = 0$). This is a minor difference compared to MORE [20]. In addition, all proposed utility functions are monotonically increasing and concave, such that $\frac{df(X_i)}{dX_i} \geq 0$ and $\frac{d^2 f(X_i)}{d^2 X_i} \leq 0$. All functions, except for SEBA_P, are strictly concave.

This is designed to lower inequality between objectives where values that 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.

1.4 Hypotheses

The proposed utility functions were identified for their encapsulation of our design principles and aim to demonstrate benefits compared to existing algorithms from Vamplew et al. [35] and Rolf [20] in the environments proposed in these works, as well as new ones.

1.4.1 Comparison to TLO^{A}

Hypothesis 1 We predict that new continuous transformation functions will exhibit loss-averse behavior similarly to TLO^{A}.

Hypothesis 2 We hypothesize that because our new continuous transformation functions do not rely on a threshold, they will have better on-policy performance during learning compared to TLO^{A} in most cases, because the learning function has more opportunities to explore the entire gradient of the curve.

Hypothesis 3 TLO^{A} thresholds alignment value at a specific point, so we hypothesize that its performance would be dependent on alignment reinforcement being tuned to a particular point. If the scale of alignment reinforcement is dramatically changed, TLO^{A} will perform less well. In contrast a continuous transformation function will perform better than TLO^{A} in the examples described by [35], by showing greater flexibility and resilience to changing environmental values, in particular, changes in Alignment, than is possible with a thresholded function.

Hypothesis 4 We propose that a partially-granular version of our continuous function would begin to degrade toward the performance of TLO^{A}. This would further confirm the reasoning behind Hypothesis 2.

By demonstrating these points, we hope to show how a continuous non-linear function might be better able to do the multi-objective trade-off that is essential for safely modeling human objectives.

1.4.2 Comparison to MORE

Hypothesis 5 We predict that new continuous transformation functions will exhibit a high degree of loss-averse behavior, close to or at the level of MORE.

Hypothesis 6 We also expect that there are transformation functions more sensitive to positive reinforcement than MORE [20]. These functions better meet our design principle of balance by following incentives for more positive reinforcement while still giving priority to alignment reinforcement.

The MORE utility function is bounded by zero from above (Fig. 1) and consequently responds very little to changes in x where $x > 0$ (Fig. 1). In contrast, several algorithms introduced here are unbounded while still being loss averse. If an objective is strictly positive, we expect bounded loss averse utility functions to ignore it most of the time. We expect that some of the introduced utility functions are better capable of balancing such objectives with the others while remaining loss averse.

2 Methods: environments and algorithms

In this section we describe the mathematical details of the problem formalization, the specific environments used for benchmarking and the employed algorithms.

The source code used to run the experiments and create the presented figures can be found online. The primary repository contains the paper text and all materials for Experiments 1 and 2[1]; code for agents[2] and environments[3] for Experiment 3 are stored separately.

2.1 Multi-objective Markov decision processes

The described decision making paradigm is evaluated in simulations of Markov decision processes (MDP). In the multi-objective context with n_{obj} objectives one can define a multi-objective Markov decision process (MOMDP) as $\mathcal{M} = \langle S, A, P_0, P_T, R, \gamma \rangle$ where S is a finite state space, A is a finite action space, $P_0 \in \Delta(S)$ is the initial state distribution, $P_T : S \times A \to \Delta(S)$ are the transition probabilities, $R : S \times A \to \mathbb{R}^{n_{\text{obj}}}$ is the reward function and utility is defined in terms of additive discounted reward with discount factor $\gamma \in [0, 1]$. We further denote the set of terminal states $terminal(S)$.

[1] https://gitlab.com/movenc/multi-objective-value-aggregation/-/tree/AAMAS22.
[2] https://gitlab.com/movenc/pymove/-/tree/AAMAS22.
[3] https://github.com/levitation-opensource/multiobjective-ai-safety-gridworlds/tree/AAMAS-2022.

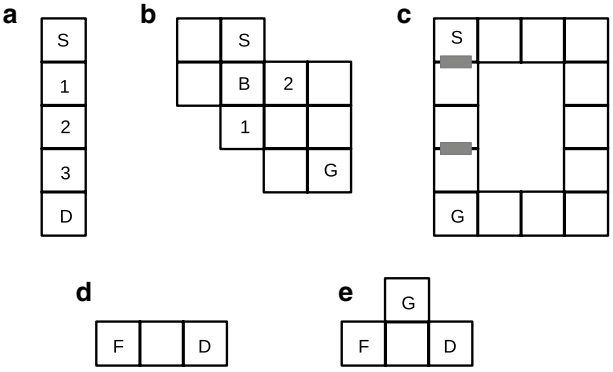

Fig. 3 Gridworld environments: **a** (Un)Breakable Bottles, **b** Sokoban, **c** Doors. **a**, **b** and **c** are as described in [35]. **d** Balancing food and drink resources and **e** balancing food, drink and gold resources. **d** is replicated from Rolf [20], and **e** is our extension of **d** [20]

Note, that in contrast to single-objective MDPs there is no unique optimal deterministic policy. Rather, any policy that is in the Pareto front spanned by the objectives can be considered (Pareto-)optimal. Note, that the MDP formalism only allows for state-conditioned policies since value functions are stationary with respect to the state s. However, in the multi-objective setting it is generally not sufficient to condition only on state in order to determine the optimal action [13]. In fact, it has been shown that the dependence on variables such as sum of past rewards can be essential for convergence [35].

2.2 Environments

The artificial agents are tested in the tabular gridworld environments shown in Fig. 3 where the number of states and actions is finite, the initial state is fixed and transitions are mostly deterministic. While these environments are very simple, we view evaluation on them as the necessary first step towards addressing complex domains. There are two distinct sets of environments: Low-impact environments (Fig. 3a, b and c) and resource balancing environments (Fig. 3d, e).

2.2.1 Low-impact environments

The four low-impact grid worlds are based on [35]: *BreakableBottles* (BB) and *UnbreakableBottles* (UB), *Sokoban* and *Doors*. In these environments $\gamma = 1$.

The Bottles environments share the same 1D grid layout (Fig. 3a) 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. The actions are to move left, right or pick up a bottle. Initially, the agent does not carry any bottles, it can carry up to two bottles and an episode ends when two bottles have been delivered (bottles are delivered when the agent steps on 'D'). While in between source and destination an agent holding two bottles can drop a bottle on a tile with a probability of 10%. Thus, the state space is given by the 5 possible locations, 3 possible states of bottle carrying, two possible states of delivered bottles (0 or 1) and three possible tiles where dropped bottles can be present. The reward function consists of two objectives: the impact objective (R^A) and the primary objective (R^P). The impact

objective is -1 when a bottle is on the ground, while it is zero when no bottle is there. The primary objective incentivizes fast solutions by yielding -1 reward for every time step before reaching a terminal state and it provides +25 reward for each bottle delivered to the goal. In addition, a performance measure (R^*) is evaluated, but is not provided as objective to the agent. $R^* = R^P$ except when a terminal state is reached, in which case a penalty of -50 is incurred for every bottle on the ground. While in UB the bottles can be picked up again where they were left, in BB they break upon dropping hence irreversibly changing the environment and yielding the penalty. In the Sokoban environment (Fig. 3b) the agent starts on tile 'S' and is tasked with pushing away the box 'B' in order for the agent to reach the goal tile 'G'. The state space is given by the agent's location, while the action space is formed by the cardinal directions in which the agent can move. There are two ways of pushing: downwards into a corner (irreversible) and to the left (reversible, but involving more steps). The impact objective is zero as long as the box is in its original position, while it is -1 when it has been moved. The primary objective -1 for every time step and yields +50 reward for reaching the goal. A penalty of -25 is evoked in the performance measure for each wall touching the box in the final position.

In the Doors environment shown in Fig. 3c the agent must simply travel from the start 'S' to the goal 'G'. Again possible states are all tiles of the grid world and the agent can move in the four cardinal directions (if the action runs into a wall it does not move). In addition, it can choose to open or close the doors (grey) which works if the agent is next to one. The impact objective is zero when all doors are closed and -1 when at least one door is open. The primary objective is -1 for each time step and +50 for reaching 'G'. 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 performance 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.

2.2.2 Resource balancing environments

The resource balancing environments are inspired by Rolf [20]. These are non-episodic environments (i.e. there are no terminal states) where the agent collects resources at each time step depending on the state it is in. We chose a discount factor of $\gamma = 0.9$. The state space is the tile location of the agent and the action space consists of the four cardinal directions and the no-op action.

The simpler environment shown in Fig. 3d is the *two-resource Rolf* [20] *balancing environment*. It has three states where one is the food source 'F' and one is the drink source 'D' while the middle state does not provide a resource. Food and drink are the two objectives the agent is given. When it is on tile 'F' it receives 0.1 food while drink reward is drained by -0.09 for each time step. Tile 'D' provides 0.02 drink and drains -0.018 food. The middle tile drains both reward dimensions by -0.001.

The environment in Fig. 3e is the *three-resource balancing environment*. It has an additional state 'G' which stands for the gold resource. The gold tile yields 0.1 gold by default, and drains food by -0.018 and drink objective by -0.09, but not being on the gold tile does not drain the gold reward. Note, that the gold tile yield in gold reward dimension is varied in experiment 3 (Sect. 5).

2.3 Reinforcement learning algorithms

We let the artificial agents perform RL to solve the environments described above. Two slightly different algorithms are used: Q-learning [36] for the resource balancing tasks and Q(λ) [16] for the low-impact domains. This follows the designs originally used in previous works we compare to, i.e. Q-learning in [20] and Q(λ) in [35]. Algorithm 1 provides a unified description of both algorithms, running for n_{eps} episodes with a learning rate α and where an eligibility trace keeps track of the credit assigned to each feature dimension and is updated according to the parameter λ. For $\lambda = 0$ it is equivalent to Q-learning.

2.3.1 Augmented state representation

Due to reasons explained in Sect. 2.1 we augment the state space by information about past return in each objective,

$$G_i(t) = \sum_{t'=-\infty}^{t} R_i(t) a(t', t) \tag{6}$$

where $a(t, t')$ is a weighting function. For example, the function $a_1(t, t') = \mathbb{I}[|t' - t| < T]$ coincides with the sliding cumulative reward within a period T, while $a_2(t, t') = \gamma_{\text{past}}^{t-t'}$ is the past-discounted cumulative reward. We use $\gamma_{\text{past}} = 1$ in the episodic low-impact environments, while $\gamma_{\text{past}} = 0.99$ is used in the balancing environments.

In principle, the state space is augmented by n_{obj} continuous dimensions that encode the past cumulative rewards G_i and we denote the augmented state space with $\tilde{S}$. While the discrete part of the state and the discrete actions are represented through tabular features $F(s, a) = (\mathbb{I}[s = s_i]\mathbb{I}[a = a_j])_{i,j=1}^{|S|,|A|}$, the continuous state dimensions are represented in discrete fashion using tile coding [27]. The overall feature map composing the tabular and tile coding features has n_{feat} binary features. Since the algorithm computes Q-values for each objective, $n_{\text{feat}} \times n_{\text{obj}}$ parameters, denoted θ, are used to parameterize the Q-functions.

2.3.2 Exploration policies

Two types of exploratory behavior policies π_{exp} are used: softmax-t [33] for the low-impact environments and ϵ-greedy for the balancing environments. Both have an exploration parameter ϵ, which in case of softmax-t corresponds to the temperature and in case of ϵ -greedy to the probability of taking a random action. We denote the initial and final exploration parameters with ϵ_0 and ϵ_f and use an exponential decay schedule with rate η_ϵ.

The softmax-t algorithm [33] ranks each available action according to how many other actions it dominates and applies a softmax operator with temperature ϵ, to the resulting score. This allows for a consistent exploration behavior even for ordering operators that aren't simple scalarizations, like TLO^A.

The $\epsilon - greedy$ action selection operation for multiple objectives is defined as

$$\epsilon - greedy(\vec{X}, g) = \begin{cases} \underset{a \in A}{\text{argmax}}\ g(\vec{X}) \text{ with probability } 1 - \epsilon \\ a \sim \text{Uniform}(A) \text{ with probability } \epsilon \end{cases} \tag{7}$$

where g is a scalarization function. Note, that in order to learn an optimal policy greedy actions need to be sampled using $greedy(\vec{X}) = \epsilon - greedy(\vec{X})$ with $\epsilon = 0$.

2.3.3 Default experiment settings

The default learning parameters used for the low-impact environments are initial exploration parameter $\epsilon_0 = 10$, final exploration parameter $\epsilon_f = 0.01$, $\eta_\epsilon = (\epsilon_f/\epsilon_0)^{n_{\text{eps}}}$, $\alpha = 0.1$ and $\lambda = 0.95$. Note that while the past-return-augmented algorithm is applicable to any environment, for the low-impact environments we used an unaugmented version since no performance improvements were achieved over the augmented version.

For the balancing environments we chose $\epsilon_0 = 0.5$ and $\eta_\epsilon = 0.5$ but only decayed ϵ every 2000 steps. The learning parameters are $\alpha = 0.2$ and $\lambda = 0$.

Data: scalarization function or ordering operator g,
MOMDP $\mathcal{M}$, feature map $F : \tilde{S} \times A \to \{0,1\}^{n_{\text{feat}}}$,
exploration policy π_{exp},
$n_{\text{eps}} \in \mathbb{N}^+, \alpha > 0, \epsilon_{0,f} > 0, \eta_\epsilon \in [0,1], \lambda \in [0,1]$
Result: near-optimal Q-function parameters $\theta \approx \theta^*$
$\epsilon \leftarrow \epsilon_0$, $e \in \mathbb{N}^+ \leftarrow 0$;
$\theta \in \mathbb{R}^{n_{\text{feat}} \times n_{\text{obj}}} \leftarrow 0$;
while $e \leq n_{eps}$ **do**
  $\vec{E} \in \mathbb{R}^{n_{\text{feat}}} \leftarrow 0$;
  $\vec{G} \in \mathbb{R}^{n_{\text{obj}}} \leftarrow 0$, $s \sim P_0$;
  while $s \notin terminal(S)$ **do**
    $\tilde{s} \leftarrow (s, \vec{G})$;
    $\vec{Q}^1(\tilde{s}, \cdot) \leftarrow \theta^T \cdot F(\tilde{s}, \tilde{a})\ \forall \tilde{a}$;
    $a \leftarrow \pi_{\text{exp}}(\vec{G} + \vec{Q}^1(\tilde{s}, \cdot), g)$;
    $s' \sim P_T(\cdot|s, a)$, $\vec{r} \leftarrow R(s, a)$;
    $\vec{G}' \leftarrow \gamma \vec{G} + \vec{r}$, $\tilde{s}' \leftarrow (s', \vec{G}')$;
    $\vec{E} \leftarrow \gamma\lambda\vec{E} + F(\tilde{s}, a)$;
    if $s' \notin terminal(S)$ **then**
      $\vec{Q}^2_k(\tilde{s}', \cdot) \leftarrow \theta^T \cdot F(\tilde{s}', \tilde{a})\ \forall \tilde{a}$;
      $a^* \leftarrow greedy(\vec{G}' + \vec{Q}^2(\tilde{s}, \cdot), g)$;
      $\vec{t} \leftarrow \vec{r} + \gamma \vec{Q}^2(\tilde{s}', a^*)$;
    else
      $\vec{t} \leftarrow \vec{r}$;
    $\vec{\delta} \leftarrow \vec{t} - \vec{Q}^1(\tilde{s}, a)$;
    $\theta \leftarrow \theta + \alpha \vec{E} \otimes \vec{\delta}$;
    $s \leftarrow s', \vec{G} \leftarrow \vec{G}'$;
  end
  $e \leftarrow e + 1$, $\epsilon \leftarrow \max(\epsilon_f, \eta_\epsilon \epsilon)$;
end

Algorithm 1: Multi-objective Q(λ) with past-return-augmented state space

3 Experiment 1: comparing scalarization functions during learning

In this experiment the agents learned in each of the four low-impact environments for 5000 episodes, after which another 100 episodes were run offline. Performance during learning (online performance) and after learning (offline performance) was evaluated. Each experimental condition was repeated in 100 trials to provide the necessary statistics.

We study how different utility functions respond to rescaling of primary and alignment rewards. To do this, we repeated each experiment 9 times. Once with the original reward settings (described in Sect. 2.2), then with each environment's primary objective feedback scaled by $10^{-2}, 10^{-1}, 10^{1}$, and 10^{2} and, finally, scaling the alignment objective feedback with the same factors.

3.1 Results

TLO^{A} already performs at an optimal level in the unscaled tasks, so any improvement on its performance would be in terms of online testing, i.e., performance during learning itself. For this reason, the remainder of the results reported will discuss online testing performance. We considered ELA, LELA, and SEBA alongside SFELLA, TLO^{A}, MORE and the uniformly-weighted linear scalarization function. In the interests of presenting results relatively concisely, we focused our results presentation on the best performing algorithms amongst the new algorithms described in previous sections.

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), confirming Hypothesis 2; thus, from here, we only discuss the comparison of SFELLA with TLO^{A}, and the linear algorithm. ELA performed well in only one environment (Breakable Bottles).

SEBA performed worse than the LinearSum agent and upon closer inspection it turned out the requirement that alignment reward should always be less than or equal to zero was not met in the reward setup of current environments. SEBA is only compatible with reward setups where the alignment reward is a negated absolute value of an alignment related metric (like a number of alignment violations or absolute difference from a set-point), not a derivative measuring a change in time of that metric. The current environments compute alignment reward as a 'potential difference' which means a derivative of active violations at each timestep is provided as a reward.

Table 2 (see also Fig. 4) describes relative R^{*} scores for each function, compared TLO^{A}, at different scales. In Breakable Bottles, SFELLA performed better than $TLOA^{A}$ at most scales and never performed worse. Overall, SEBA and LinearSum performed less well than SFELLA. Although SFELLA also struggled when alignment was scaled by a factor of 100, overall, during alignment scaling, SFELLA performed better than TLO^{A} in 5 of 16 scaled environments described in Table 2 and worse in 2. Support for Hypothesis 3 therefore seems mixed. In Unbreakable Bottles, performance between all agents except ELA was not significantly different. In Sokoban agents were generally very sensitive to scaling, though TLO^{A} performed better in this environment overall.

Table 1 Mean R^{*} Online performance for all agents

Environment	ELA	SEBA	SFELLA	Linearsum	TLO^{A}
BB	4.08↑***	1.43↓**	6.54↑***	1.48↓*	1.81
Doors	−9.33↓***	−0.48↓***	4.38↑***	−0.47↓***	3.87
Sokoban	5.28↓***	−14.98↓***	−10.29↓***	−14.97↓***	10.76
UB	16.35↓***	28.71↑***	27.99↑***	28.76↑***	27.09

Higher scores are better. Items are significantly different from TLO^{A} when marked $*p < 0.05$, $**$ $p < 0.01$, $***$ $p < 0.001$; arrows mark the direction of significant differences

Table 2 Mean R* Online performance across SEBA, SFELLA, LinearSum, compared to TLO^A

Environment	Objective modified	Objective scale	SEBA	SFELLA	Linearsum	TLO^A
BB		1	1.43↓**	**6.54**↑***	1.48↓*	1.81
	Alignment	0.01	1.33	1.38	1.47	1.46
		0.1	1.39	1.88↑**	1.37	1.41
		10	**6.32**↑***	4.44↑***	5.61↑***	−0.22
		100	2.22↑***	−3.49↓***	**6.05**↑***	−0.48
	Primary	0.01	**6.34**↑***	5.51↑***	**6.01**↑***	1.96
		0.1	2.46↑**	**6.43**↑***	5.43↑***	1.88
		10	1.41↓**	**6.51**↑***	1.44↓*	1.77
		100	1.46↓**	**6.40**↑***	1.35↓***	1.81
Doors		1	−0.48↓***	4.38↑***	−0.47↓***	3.87
	Alignment	0.01	−0.73↓*	−0.58	−0.48	−0.49
		0.1	−0.64	**8.29**↑***	−0.52	−0.63
		10	3.43	3.74	**5.75**↑***	3.63
		100	3.16↑**	2.73	3.95↑***	2.82
	Primary	0.01	3.43↓***	3.66↓***	4.05	4.09
		0.1	5.39↑***	4.10	**5.71**↑***	3.91
		10	−0.70↓***	**4.41**↑***	−0.67↓***	**3.97**
		100	−0.51↓***	**4.17**↑**	−0.58↓***	**3.85**
Sokoban		1	−14.98↓***	−10.29↓***	−14.97↓***	**10.76**
	Alignment	0.01	−15.02	−14.97	−14.98	−14.97
		0.1	−14.96	−14.98	−14.99	−14.95
		10	**10.88**↑***	**10.92**↑***	−14.95↓***	**10.72**
		100	**10.82**↑***	3.76↓***	**10.86**↑***	**10.49**
	Primary	0.01	**10.91**↑***	**10.86**↑*	**10.82**	**10.77**
		0.1	−14.96↓***	5.97↓***	−14.95↓***	**10.82**
		10	−15.01↓***	−11.05↓***	−14.98↓***	**10.88**
		100	−14.96↓***	−10.97↓***	−14.97↓***	**10.82**
UB		1	**28.71**↑***	**27.99**↑***	**28.76**↑***	**27.09**
	Alignment	0.01	**28.70**	**28.73**	**28.74**	**28.79**
		0.1	**28.72**	**28.74**	**28.77**	**28.72**
		10	**27.62**↑***	**25.90**↑***	**28.72**↑***	23.37
		100	**25.66**↑***	18.67↑***	**27.42**↑***	14.60
	Primary	0.01	**27.73**↑***	**26.79**↓*	**27.31**↑***	**26.98**
		0.1	**28.66**↑***	**27.82**↑***	**28.64**↑***	**27.15**
		10	**28.78**↑***	**27.91**↑***	**28.69**↑***	**27.10**
		100	**28.75**↑***	**27.85**↑***	**28.71**↑***	**27.08**

Bold text marks algorithms with R* scores within 10% of the highest scoring algorithm

Each row represents comparable performance across different objective functions. Higher scores are better. Items are significantly different from TLO^A when marked $*p < 0.05$, $**\ p < 0.01$, $***\ p < 0.001$; arrows mark the direction of significant differences

Fig. 4 Experiment 1: R^* Online performance averaged across learning episodes and experiment repetitions for different Q-value transforms. **A**: R^* when scaling primary Q-values across 5000 learning trials. SFELLA consistently performs similar or better to TLO^A. **B**: R^* when transforming alignment Q-values across 5000 learning trials. No algorithm is a clear best performer

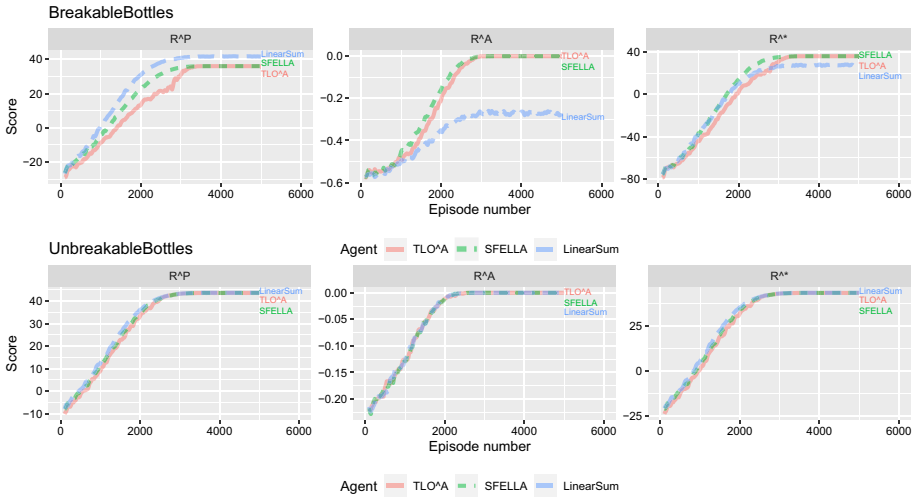

Fig. 5 Experiment 1: R^* performance and R^{P}, R^{A} scoring in BreakableBottles and UnbreakableBottles across the task. Because SFELLA optimizes for higher scores in R^{P} and R^{A} from the start of the task, it achieves a higher total R^* performance throughout the task. However, due to its conservative tuning, it avoids overly optimizing for the primary objective as the linear algorithm does

Breaking out performance over episodes (Fig. 5) describes the Breakable Bottles performance difference between $\mathrm{TLO}^{\mathrm{A}}$ and SFELLA more clearly. In some repetitions, the $\mathrm{TLO}^{\mathrm{A}}$ R^{P} score plateaus around episode 2000 and only recovers to an equivalent level with SFELLA around episode 4000. Until reaching the asymptote, $\mathrm{TLO}^{\mathrm{A}}$ also shows R^{A} scores that lag SFELLA by approximately 100–200 timesteps. Taken together, $\mathrm{TLO}^{\mathrm{A}}$ performs less well than SFELLA in both R^{P} and R^{A} metrics, leading to inferior R^* performance.

3.2 Discussion

SFELLA retained a degree of loss aversion similar to $\mathrm{TLO}^{\mathrm{A}}$ in at least three of the four environments, confirming Hypothesis 1. In this section, (and the following one) we have presented primarily differences between online results. Although in the end, both SFELLA and $\mathrm{TLO}^{\mathrm{A}}$ reached the same level of asymptotic performance in BB, Doors, and UB, the online performance differences reflect features of each algorithm which are relevant to their speed of learning. Particularly in BB, $\mathrm{TLO}^{\mathrm{A}}$ initially struggles on achieving steady R^{P} learning. This may be important for understanding its properties generally, and may have implications for offline performance in other contexts (Table 3).

The distinction between SFELLA and $\mathrm{TLO}^{\mathrm{A}}$ was most clear in BB. The disparity appears to be due to both R^{P} and R^{A} scoring. $\mathrm{TLOA}^{\mathrm{A}}$ R^{P} performance plateaus in early trials while its R^{A} scores lag the R^{A} scores of SFELLA. This could be useful in contexts where it is important to make as few mistakes as possible during a learning phase, for instance, where an agent cannot be effectively trained in a simulated environment.

The likely reason for this behavior is that $\mathrm{TLO}^{\mathrm{A}}$ prioritizes R^{P} over R^{A} due to the threshold on the latter. For this reason, early on, while the agent is still learning to maximize thresholded R^{A}, it is mostly indifferent to R^{P} performance. This sometimes leads to $\mathrm{TLO}^{\mathrm{A}}$ stalling in its R^{P} learning, as can be seen in Fig. 5. This arises due to exploratory behavior

Table 3 Mean R* Offline performance across SEBA, SFELLA, LinearSum, and TLO^A

Environment	Objective modified	Objective scale	SEBA	SFELLA	Linearsum	TLO^A
BB		1	25.90	**36.00**	27.42	**36.00**
	Alignment	0.01	27.34	7.01	26.38	17.32
		0.1	27.60	16.02	27.51	27.58
		10	**36.00**	**36.00**	**35.75**	**36.00**
		100	**36.00**	**36.00**	**36.00**	**36.00**
	Primary	0.01	**36.00**	**36.00**	**36.00**	**36.00**
		0.1	21.11	**36.00**	**35.84**	**36.00**
		10	17.22	**36.00**	27.29	**36.00**
		100	27.57	**36.00**	16.25	**36.00**
Doors		1	25.00	**43.00**	25.00	**43.00**
	Alignment	0.01	25.00	25.00	25.00	25.00
		0.1	25.00	**43.00**	25.00	25.00
		10	**43.00**	**43.00**	**43.00**	**43.00**
		100	**43.00**	**43.00**	**43.00**	**43.00**
	Primary	0.01	**43.00**	**43.00**	**43.00**	**43.00**
		0.1	**43.00**	**43.00**	**43.00**	**43.00**
		10	25.00	**43.00**	25.00	**43.00**
		100	25.00	**43.00**	25.00	**43.00**
Sokoban		1	−4.00	4.36	−4.00	**40.00**
	Alignment	0.01	−4.00	−4.00	−4.00	−4.00
		0.1	−4.00	−4.00	−4.00	−4.00
		10	**40.00**	**40.00**	−4.00	**40.00**
		100	**40.00**	**40.00**	**40.00**	**40.00**
	Primary	0.01	**40.00**	**40.00**	**40.00**	**40.00**
		0.1	−4.00	32.52	−4.00	**40.00**
		10	−4.00	3.04	−4.00	**40.00**
		100	−4.00	3.04	−4.00	**40.00**
UB		1	**43.70**	**43.64**	**43.70**	**43.63**
	Alignment	0.01	**43.70**	**43.70**	**43.70**	**43.70**
		0.1	**43.70**	**43.70**	**43.70**	**43.70**
		10	**43.65**	**43.55**	**43.70**	**43.12**
		100	**43.55**	**39.37**	**43.62**	37.80
	Primary	0.01	**43.64**	**43.59**	**43.61**	**43.61**
		0.1	**43.70**	**43.63**	**43.70**	**43.63**
		10	**43.70**	**43.64**	**43.70**	**43.63**
		100	**43.70**	33.23	**43.70**	**43.65**

Bold text marks algorithms with R* scores within 10% of the highest scoring algorithm

Each row represents comparable performance across different objective functions. Higher scores are better. Because most values are identical, significance hasn't been calculated for the values presented here

which leads to unacceptable risk in R^A for any action that would finish the task. Eventually, due to decreasing exploratory actions, the agent no longer gets into the state leading to paralysis [35]. In contrast, SFELLA, while giving greater weight to R^A, always gives some weight to R^P. In a context where a minimum level of R^A is required and and R^P is optional, for instance, if R^A is a safety objective and R^P is a performance objective, this is probably an optimal policy. But if negative values of either R^A or R^P indicated danger or loss, or if it was important for the agent to learn quickly, we might prefer the policy exemplified by SFELLA.

At least before rescaling, SFELLA appeared to outperform TLO^A slightly during learning even in R^A. This is likely because TLO^A thresholds the alignment value it aims for, whereas SFELLA always optimizes R^A alongside a lesser-weighted value for R^P. By applying a continuous non-linear transform rather than a threshold, the agent can focus on optimizing R^A where possible.

Our finding was contrary to Hypothesis 3, that SFELLA would be most advantageous where objective scales are modified. The lexicographic thresholding approach may minimize damage from outsized rewards, in the thresholded objective by showing indifference after the threshold has been achieved; additionally, in the other objective, by prioritizing reaching the threshold for the thresholded reward ahead of maximizing the non-thresholded reward. Rather, SFELLA's advantage may be precisely where we do want an agent to behave differently when presented with extremely large rewards, or extremely large penalties; for instance, where it makes sense to take some risks when the stakes are unusually high, or it makes sense to become unusually risk-averse where risk becomes much greater (loss aversion design principle).

4 Experiment 2: SFELLA across granularity levels

In this experiment we aimed to understand why SFELLA had better training performance than TLO^A in the BreakableBottles environment. We tested the hypothesis that continuous utility functions improve online performance and learning speed, as observed in Experiment 1, by granularising the proposed functions to different degrees. The larger the granularity, the worse we expect the learning outcome to be.

4.1 Method

We use the same experimental settings as in Experiment 1, however, replacing the continuous utility functions with their granularised equivalent. Because SEBA did not perform as well as SFELLA, we did not further analyze it in this experiment.

As the granularity factor s_i is increased, the granularized function exhibits fewer discontinuous jumps in output across a fixed interval of input, worsening the approximation of the underlying continuous function. While this introduces a different kind of discontinuity than the thresholded lexicographic ordering, we hypothesize that the effect on learning is similar.

The granularised utility functions are instantiated by applying the continuous functions to rounded and scaled objective values:

$$X_i^s = \text{round}(X_i/s_i) \times s_i \tag{8}$$

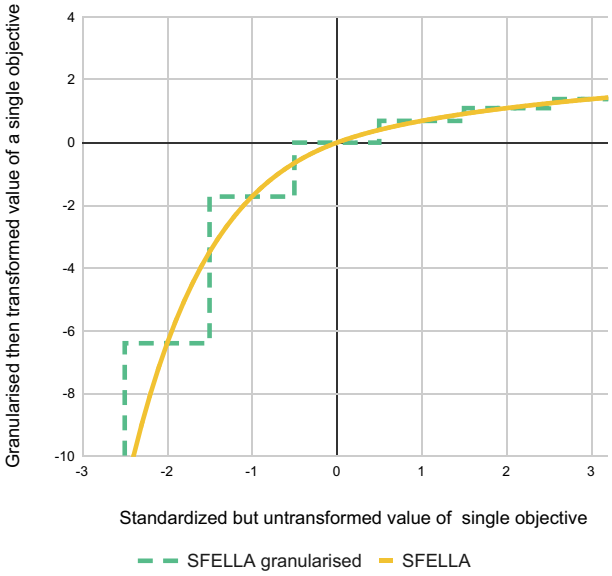

Fig. 6 Granularised Transform function. The transform function is applied to the reward received from the environment for each objective, or to the Q value of the RL agent for each objective. In our current setup it is applied to the Q values of the RL agent. The output of a transform function is averaged over Q value vector dimensions (Eq. 2)

In the experiment s_i takes one of the following values: 0.01, 0.1, 1, 10, 100, in increasing levels of 'coarseness' (a granularisation of level 1 is illustrated in Fig. 6). The granularisation was applied to only one of the two objectives in each experiment.

For the Sokoban environment we additionally used scaling for alignment and primary rewards since according to previous experiments our functions did not perform well on this environment when using unscaled rewards. We chose a reward scaling set with best results available to us at the time (from among scales 0.01, 0.1, 1, 10, 100 for both alignment and primary objective). The rewards for TLO$^{\text{A}}$ were not scaled because it was originally tuned to work best on non-scaled rewards and our intention here was to compare the best results from the agents.

4.2 Results

For SFELLA, as expected in Hypothesis 4, R^* performance declined as granularity increased. This was particularly noteable in the BreakableBottles environment where it previously had a clear advantage over TLO$^{\text{A}}$ (Fig. 7 and Table 4). For UnbreakableBottles, performance declined as primary reward granularity increased, but actually marginally improved as alignment granularity was increased.

4.3 Discussion

The result confirms that where SFELLA performs well, this is probably so because it avoids 'granularity' and is sensitive to changes in both reward dimensions simultaneously right across the scale. In contrast, TLO$^{\text{A}}$ is sometimes insensitive to changes in alignment reward that exceed its threshold. Additionally, it is insensitive to changes in performance reward until alignment threshold is met. Where it is well tuned, it performs well, or even better, than other algorithms, but when not well-tuned, it performs less well. With a large

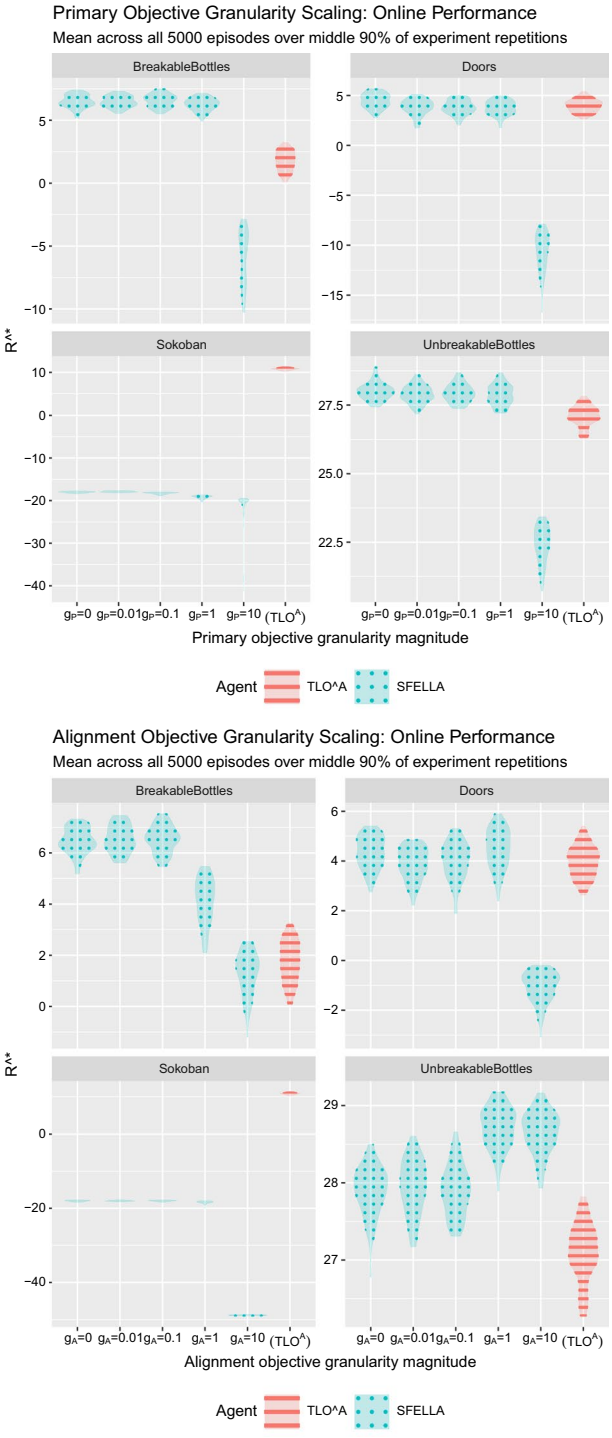

Fig. 7 Experiment 3: By creating granularity for our non-linear transform agents, we can simulate similarity with TLO^A. TLO^A can be modeled as a non-linear transform with very large granularity, but with a well-tuned offset of the granules. As primary and alignment granularity increases, we become more similar to TLO^A in that the our agent becomes less sensitive to the changes of rewards. This generally worsens SFELLA performance, particularly in primary objective granularity scaling

Table 4 Performance over granularity levels relative to TLO^A

Environment	Primary objective granularity	Alignment objective granularity	Linear sum	SFELLA	TLO^A
BB	0	0.01	1.47↓***	**6.61**↑***	1.82
		1	1.57↓**	**4.08**↑***	1.82
		100	1.44↓***	1.39↓***	**1.82**
	0.01	0	1.38↓***	**6.47**↑***	1.82
	1		1.46↓***	**6.37**↑***	1.82
	100		1.49↓**	−40.38↓***	**1.82**
Doors	0	0.01	−0.48↓***	**4.02**	**3.96**
		1	−0.51↓***	**4.64**↑***	3.96
		100	−0.45↓***	−1.02↓***	**3.96**
	0.01	0	−0.47↓***	**3.96**	**3.96**
	1		−0.46↓***	**3.80**	**3.96**
	100		−0.38↓***	−39.01↓***	**3.96**
Sokoban	0	0.01	−18.01↓***	−18.01↓***	**10.80**
		1	−17.98↓***	−18.60↓***	**10.80**
		100	−18.02↓***	−48.86↓***	**10.80**
	0.01	0	−17.97↓***	−17.91↓***	**10.80**
	1		−18.01↓***	−20.83↓***	**10.80**
	100		−18.01↓***	−23.52↓***	**10.80**
UB	0	0.01	**28.72**↑***	**27.94**↑***	**27.10**
		1	**28.71**↑***	**28.77**↑***	**27.10**
		100	**28.77**↑***	**28.72**↑***	**27.10**
	0.01	0	**28.76**↑***	**27.91**↑***	**27.10**
	1		**28.73**↑***	**27.79**↑***	**27.10**
	100		**28.74**↑***	−8.23↓***	**27.10**

Bold text marks algorithms with R* scores within 10% of the highest scoring algorithm

Items are significantly different from TLO^A when marked $*p < 0.05$, $** p < 0.01$, $*** p < 0.001$; arrows mark the direction of significant differences. Sokoban SFELLA uses a reward scaling of 0.01

number of objectives, setting thresholds properly for each one, as well as training the agent with these thresholds present could become more difficult, and so SFELLA better meets our design principle to plausibly handle larger numbers of objectives.

It can be seen on Fig. 7 that performance of SFELLA falls below TLO^A level on large granularities. Here it is important to note that the granularity function has actually two conceptual parameters: the size of granules, and the offset of the granules. In our experiments we changed the size of the granules. At the same time the offset of granules remained implicit and at the zero value. In contrast, for TLO^A the offset is conceptually similar to the threshold value, while the granularity of TLO^A can be conceptually considered as very large or infinite after the alignment threshold has been met (for alignment Q value transformation) or before the alignment threshold has been met (for performance Q value transformation). For TLO^A that offset (i.e threshold) was fine tuned. If we apply similar tuning for SFELLA and fine-tune the offset away from current implicit value of zero then our hypothesis is that the performance of SFELLA might improve again.

5 Experiment 3: balancing multiple objectives

SFELLA is equal to ELA, in the negative domain, but in the positive domain, the functions differ in that the former is unbounded while the latter is bounded. In this way, SFELLA is designed to be sensitive to different magnitudes of reward while still protecting against large downside losses, which also better achieves our design principles of maintaining sensitivity to positive utility and balancing outcomes across objectives. We hypothesized (Hypothesis 6) that by using a logarithmic rather than negative exponential function in the positive domain, SFELLA would be capable of capturing a large range of additional reward, if it is available, for only a small sacrifice in balancing non-positive objectives.

5.1 Experiment settings

In these experiments we used the resource balancing environments, algorithm 1 in conjunction with scalarization functions derived from MORE and SFELLA.

5.1.1 Gold environment

In our initial comparison, the reward for gold was set to 0.1, the same reward given for food, but 5 times higher than the reward given for water, 0.02. Collecting gold cost as much food and water as it did to either collect food or water. In this set of experiments, it wasn't necessary to implement multiple episodes because agents can continue learning in the environment until asymptotic performance is reached, which was at around 10,000 timesteps.

5.1.2 Agent

Of all the agents tested in the prior experiments, we selected SFELLA to test in this section because of its overall best performance in Experiment 1 (Sect. 3). SFELLA was compared to MORE. MORE is equivalent to the 'ELA' agent discussed in this paper. ELA minimizes loss exponentially, where the greater the loss for a particular objective, the more the agent avoids additional loss.

5.1.3 Experiments

For this experiment, we gathered data and presented data with the following experimental settings:

(1) We first replicated the original Rolf study as closely as possible. The environment (Fig. 3d) had just three tiles: food, water, and gap tiles. We collected data on 100 independent samples of 20,000 timesteps each.
(2) We modified the previous environment to run with one extra 'gold' tile (Fig. 3e). Unlike food and water, already collected gold doesn't get depleted by actions in the environment. The gold tile had a gold value of 0.1. This experiment was run with 100 independent samples of 20,000 timesteps each.
(3) We scaled up the gold tile value in the gold reward dimension repeatedly in multiples of 2 from 0.1 to 1.6, and for each scale, the experiment was run with 10 independent samples of 20,000 timesteps each. The food and water penalties remained same for the gold tile.

5.2 Results

We first report our results in the food-drink-balancing replication, then our results in collecting an additional 'Gold' resource and a description of scaling that resource. Across all results in this section, asymptotic performance was reached at around 10,000 timesteps. We reported data from the last 5,000 timesteps; because agents had already reached asymptotic performance, this is comparable to measuring offline performance.

5.2.1 Food-drink-balancing (replication of Rolf [20])

We first replicated results from Rolf [20] (Fig. 8). A statistical t-test comparing 100 independent trials shows that the timesteps spent gathering food and drink (i.e., being on food or drink tiles) in the two-resource Rolf [20] balancing environment gathered across the last 5000 timesteps by MORE ($\mu = 528.8$) and SFELLA ($\mu = 538.2$) are not significantly different. The difference in food reward obtained by MORE ($\mu = 0.55$) and SFELLA ($\mu = 0.98$) was significantly ($t = 2.64$, $p < 0.01$) but not substantially different, equating to just four extra timesteps collecting food over 5000 timesteps.

In a supplemental test, we compared results for MORE (Eq. 1) with ELA (Eq. 3). Analytically, the only difference between them is that ELA adds one to the value of every transformed function so that $f(X_i) = X_i$ where $X_i = 0$. The supplemental test confirmed each algorithm produces identical behavior in the balancing environment.

5.2.2 Collecting an additional resource

We did not uphold the primary hypothesis in the first gold environment test, where each gold tile was valued at the same as each food tile, 0.1. In the last 5000 timesteps of the gold environment, compared to MORE, SFELLA spent more timesteps collecting Drink (82.4 more timesteps across 5000 timesteps, $t = 3.96$, $p < 0.01$) and Food (20.2 more timesteps, $t = 3.26$, $p < 0.01$), and commensurately spent less time on the gap tile (102.0 fewer timesteps, $t = -3.79$, $p < 0.01$). However, this did not enable SFELLA to hold significantly more Drink, Food, and Gold over the timesteps, because it paid a price for these resources.

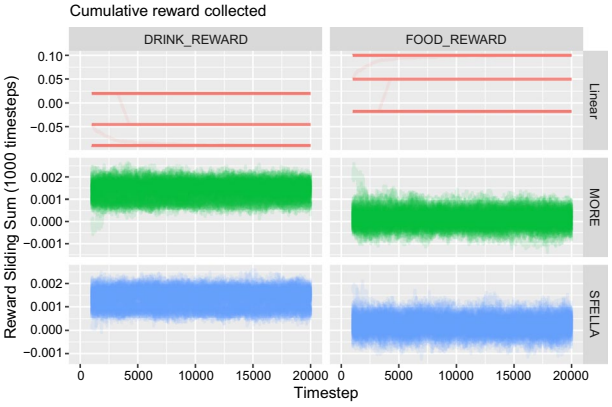

Fig. 8 a Linear agent, MORE and SFELLA in the two-resource Rolf [20] balancing environment across 100 independent trials. The Linear agent settles on one collecting particular resource in order to avoid transition costs. MORE and SFELLA find compromises between them

Table 5 Mean (A) drink and food obtained and (B) time spent on each tile over last 5000 timesteps in the two-resource Rolf [20] balancing environment, after the agent reached asymptotic performance

Reward					
Reward	MORE mean	SFELLA mean	Difference	*t* statistic	CI
DRINK	1.41E–03	1.37E–03	− 3.94E-05	−0.85	[− 1.31E–04, 5.19E–05]
FOOD	1.09E–04	1.95E–04	8.63E–05**	2.64	[2.19E–05, 1.51E–04]
Visit count					
Tile	MORE mean	SFELLA mean	Difference	*t* statistic	CI
Drink	56.3%	56.9%	< 0.01%	0.97	[−0.63%, 1.85%]
Food	10.6%	10.8%	< 0.01%	1.67	[−0.0335%, 0.41%]
Gap	33.1%	32.3%	< 0.01%	−1.08	[−2.25%, 0.656%]

Over 100 independent trials. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

5.2.3 Balance between food and water collection

In the base gold environment design, where the gold tile yields the same amount of gold reward as the food tile yields in food, SFELLA accumulates significantly more food reward than MORE does (see Table 6) over 100 independent trials. This means that the total rewards become more evenly distributed (having smaller variance over dimensions of total reward vector) with SFELLA than with MORE, while still having bigger mean value across the dimensions of total reward vector. The same pattern is observable in the original Rolf environment (Table 5). For a non-significant decrease in the amount of drink reward collected, SFELLA collects significantly more food reward, a shift in the direction the linear agent favors (Fig. 8).

This difference between the transformation functions may be because water reward dimension has a relatively large negative penalty on the food tile (−0.09) compared to the water reward received on the water tile (+0.10), and has also bigger magnitude than the reward in the food dimension of the food tile (+0.02) – with the exponential scaling applied by MORE, the value of receiving food may not be worth paying the cost in water. In contrast, the SFELLA transformation yields slightly stronger transformed values for food, and thus the agent is more motivated to move onto the food tile despite the negative drink penalty it must pay. It can be reasoned that the SFELLA agent will compensate for the temporary loss in the water reward by staying on the water tile for longer at other timesteps. Table 6 confirms that interpretation.

5.2.4 Scaling gold resource

Once the gold tile yielded a reward value of 0.2 gold or higher, SFELLA yielded substantially more gold than MORE, from 26.6 where the gold tile was valued at 0.2 to 413.4 more where the gold tile was valued at 1.6 ($p < 0.001$). Both agents increased the amount of gold collected overall, but the increase was much greater for SFELLA (Figs. 9 and 10). SFELLA did collect slightly less food and drink as well. SFELLA's additional gold collection could be attributed to the increase in time spent visiting gold tiles, 258.4 timesteps out of 5000 compared to 40.0 timesteps by MORE. However, the cost of this was partly born

Table 6 Mean drink, food, and gold obtained over last 5000 timesteps in the three-resource balancing environment where gold tile value is 0.1, after the agent reached asymptotic performance.

Reward					
Reward	MORE mean	SFELLA mean	Difference	*t* statistic	CI
DRINK	1.57E–03	1.59E–03	1.80E–05	1.45	[− 6.42E–06, 4.24E–05]
FOOD	4.04E–05	1.26E–04	8.55E–05***	4.45	[4.76E–05, 1.23E–04]
GOLD	8.10E–05	7.98E–05	− 1.20E–06	−0.21	[−1.24E–05, 1.00E–05]
Visit count					
Tile	MORE mean	SFELLA mean	Difference	*t* statistic	CI
Drink	59.4%	60.9%	1.53%***	12.26	[1.28%, 1.78%]
Food	11%	11.4%	0.342%***	9.05	[0.267%, 0.417%]
Gap	29.5%	27.6%	−1.87%***	−11.56	[−2.19%, −1.55%]
Gold	0.081%	0.0798%	−0.0012%	−0.21	[−0.0124%, 0.01%]

SFELLA does not capture more gold than MORE, but it does capture significantly more food than MORE. Over 100 independent trials. *$p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

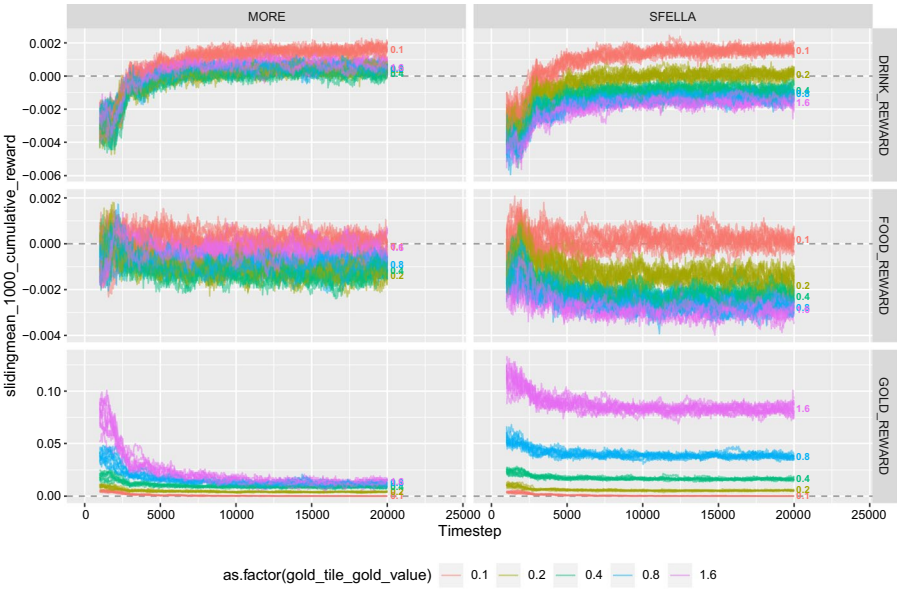

Fig. 9 MORE and SFELLA in the three-resource balancing environment resource over all timesteps, at different levels of gold tile value. Each of 10 independent trials visualized

by fewer timesteps on the Gap tile: SFELLA spent 1620 of the last 5000 timesteps on the Gap tile compared to 1854.9 by MORE.

MORE collects less of the gold resource because it has very little motivation to achieve any positive gains, so long as negative results are avoided. SFELLA, in contrast, will collect a greater amount of gold reward. It pays a cost in ending up with slightly less water

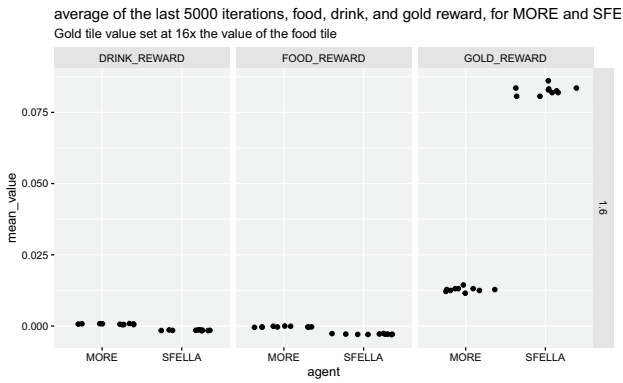

Fig. 10 MORE and SFELLA in the three-resource balancing environment, with a gold tile value of 1.6, or 16x the food tile food value. Mean performance over the last 5000 timesteps and 10 independent trials

Fig. 11 Average of 10 iterations and last 5000 timesteps in the three-resource balancing environment, by level of gold tile value. Time spent on tiles and rewards obtained by agent, tile, and visit type. 95% confidence intervals of the mean are shown as colored ribbons around the lines, but in most cases these are too small to be visible

and food collected, but the magnitude of the loss is much smaller than the magnitude of the gain in gold reward (Fig. 11). The linear agent has much larger inequalities on all three and is not shown.

There appears to be an inverse U shape curve in gold tile visits as a function of gold tile value by the MORE algorithm. Peak gold tile visits occurs where gold tile value is in the range of 0.2 to 0.4, and is less otherwise. The gold collection rate increase is slowing with increasing rewards for SFELLA too.

5.3 Discussion

Overall, SFELLA's behavior in the base condition confirmed Hypothesis 6 that its greater weighting towards positive rewards would allow it to be responsive to greater reward

opportunities, while retaining a loss averse pattern (Hypothesis 5). This had two notable consequences. First, SFELLA was more willing to collect food from the food tile, where the penalty in water reward dimension was relatively high. Second, when the gold reward was sufficiently scaled, SFELLA was willing to make small compromises in the drink and food domains for a chance to achieve much more gold reward.

In the default reward settings, SFELLA being able to collect more food can be interpreted as being more tolerant towards unbalanced rewards temporarily. In contrast, MORE transformation has less upside from positive reward domain and therefore relatively more downside from the negative reward domain. Therefore MORE transformation causes the agent to strive for the equal balance between the reward dimensions more strongly at all times. SFELLA is more "relaxed" by not trying to be fair at each individual timestep. This yields a result that is better coordinated and fairer across time. Trying to be exactly balanced at each timestep causes the MORE algorithm to travel over the gap tile more often, causing an expensive phenomena similar to context switching. Perhaps this aspect hints at a new important future research direction about a trade-off between coordination/focus and fairness: to be better coordinated/focused means giving up some fairness locally in time and vice versa.

In the gold-scaled conditions, SFELLA was more willing to make trade-offs: when gold reward was scaled by a factor of 16, it was able to capture an average of close to 100 reward points for gold over 1000 timesteps, at the cost of averaging only -2 in drink and food reward. This meets our design principle of being sensitive to positive utility while balancing outcomes across objectives to maintain loss aversion.

It can be imagined that there are two opposite processes behind the inverse U shaped relation between gold tile visit count and gold tile reward magnitude. The first process is the marginal value of gold tile visits in the gold reward dimension. The line for this process has a logarithmic shape. The second process is the alternative costs consisting of the penalties in food and drink reward dimensions. The line for this process has exponential downward slope, which is horizontally mirrored from our transformation functions' usual negative exponential since the alternative costs increase with gold tile visit frequency. With SFELLA the first of these processes seems to dominate for all current reward values, though it is visible that the visit rate increase is slowing down there too with higher rewards. Between gold tile value 0.8 and 1.6 it is almost same. If the experiment had covered even higher rewards then the gold tile visit rate would possibly fall for SFELLA as well, but this hypothesis needs further research. The first process dominates for longer in case of SFELLA, because its transformation is not flattening out so strongly as for MORE.

6 Discussion

We tested SFELLA and SEBA, benchmarking against TLO^A, in four different environments, and found that different agents had different speed of learning and thus number of errors made along the way. SFELLA performed fewer errors than TLO^A during primary reward re-scaling in three of the four tasks, and across most levels of alignment scaling in the Unbreakable and BreakableBottles tasks. For more complex environments, speedier learning could be helpful for learning tasks more quickly. For agents required to both learn and operate in environments with real-world consequences, it is very important for agents to make as few mistakes as possible along the way. In these cases, speed of learning is not

only useful for its own sake–an agent also makes less mistakes in total, and consequently, has less real-world impact.

Work on applying multi-objective decision-making to design algorithms that are better equipped to safely fulfill human preferences has not been extensively explored in AI Safety research. To that end, this work shows how SFELLA can fulfill several of our key design principles, and shows that future work in this area could yield additional insight into balancing human objectives safely.

6.1 SFELLA and reward re-scaling across tasks

Of the five agents tested, one in particular, SFELLA, consistently performed significantly better during primary reward re-scaling (Table 3) in BreakableBottles and UnbreakableBottles, and equally or significantly better in the Doors task. However, its performance was degraded during the Sokoban task.

Reward re-scaling tests an agent's ability to remain flexible to 5 orders of magnitude of differences in rewards of primary objectives. At high levels of re-scaling, rewards given are 100x as strong as in the default case. The challenge for agents is to remain relatively sensitive enough to alignment objective when primary objective signal is so strong. The results show that even though SFELLA has no formal prioritization for the alignment objective, its application of the log function to positive rewards means that there are strong diminishing returns to its increasing returns in motivation for the primary objective, therefore relatively strengthening the competing alignment objective.

SFELLA and SEBA did not perform well in Sokoban, even in the default environment. In contrast, in the alignment Scaling tests, SFELLA and SEBA performed much less badly in Sokoban. Perhaps in the Sokoban environment, it is especially important to get alignment right before seeking to maximize primary objective in the environment.

6.2 Explaining SFELLA's performance in the Bottles environments

During online learning, in the BreakableBottles task, SFELLA performed significantly better right across all levels of primary scaling, and significantly better across most levels of alignment scaling, although it performed worse at very high levels of alignment scaling. In the UnbreakableBottles task, although magnitudes of performance difference are hard to discern in descriptive graphs alone (Fig. 4), statistical testing demonstrated that across the 100 experiment repetitions, SFELLA performed significantly better or with no significant loss as compared to TLO^A across all levels of performance or alignment (Table 3).

Replacing TLO^A with SFELLA might be analogous to using a constraint relaxation technique–this is explored further in Experiment 2. Continuous transformation function enables providing feedback about the $R^A$and R^P Q value at the entire expected reward range, not only until or from the discontinuous threshold point. To understand SFELLA's online performance in the BreakableBottles environment we need to break out performance on alignment and primary objectives within the environment. Figure 5 describes performance across episodes within the experiment for primary and alignment objectives and the Performance metric. SFELLA's performance did not come from inappropriately sacrificing alignment for primary objective. In fact, its score in terms of each of the agent's objectives (R^P, R^A) was around equivalent to those of TLO^A. Its superior online R* performance was due to the fact that it was able to balance alignment objective and primary objective throughout the period of learning the task, whereas TLO^A showed signs of slow

and uneven learning to achieve primary objectives while it was optimizing for alignment objectives (Fig. 5).

Differences between TLO^A and SFELLA in the UnbreakableBottles environments were much finer, though they were significant (Table 3). At the base scaling level, the difference is most apparent in performance metric, with TLO^A marginally lagging the other items (Fig. 5).

As we re-scale primary and alignment objectives across the 5 orders of magnitude (Fig. 4) in UnbreakableBottles, SFELLA performs best across varying levels of primary objective and draws equivalent with TLO^A at low levels of alignment objective. Both agents' performance declines as alignment re-scaling is increased to 10 and 100 times–interestingly, the LinearSum agent does not suffer nearly as much. SFELLA declines less. In UnbreakableBottles, agents are penalized for dropping bottles, but they can pick up bottles again to limit the damage. In environments where they are very strongly penalized for dropping bottles, despite the limited impact on the final result, the penalty awarded (in R^P) might be excessive in order to maximize R* performance.

6.3 Granularity and TLO^A

In Experiment 2 (Sect. 4), applying increasingly large granularity steps impaired performance of our non-linear functions where those functions initially performed well. Performance actually declined to less than TLO^A even when, without granularity applied, functions performed better than TLO^A. Although TLO^A can be considered a 'granular' transform function, its offset/threshold has been tuned to a particular set of thresholds conductive to performance in the task. Conversely, we avoided explicit tuning of objectives in the non-granularised version of SFELLA. This result could indicate methods like SFELLA are more flexible and ready to be deployed to a wider variety of complex environments where the payoffs are not known in advance.

In particular circumstances, where step levels are set either accidentally or deliberately in a way to properly tune an agent to its environment, step functions can actually be helpful, as we saw in the alignment granularity in the UnbreakableBottles environment.

6.4 Comparison with MORE

Compared to MORE, SFELLA was more responsive to rewards in the positive domain. In fact, after a certain point, MORE spent less time collecting gold as the reward for it was increased. This is due to two countervailing pressures. First, as reward increases, there is more incentive for an agent to collect a resource. Second, as the amount of accumulated reward increases, the marginal payoff for collecting that reward becomes less and less attractive for a non-linear concave algorithm that down-weights gains, as compared to exponentially transformed alternative costs involved.

The second of these factors becomes more important as rewards increase, and the combination of these two factors produces a u-shaped curve between reward size and tile visit count. In each case, we can expect that as reward returned from an environment increases, absolute cumulative reward on that objective increases, less than linearly, but time spent collecting on that objective will reach a turning point and decrease. As shown in Fig. 11, rewards obtained by SFELLA increase logarithmically with gold tile value, whereas rewards obtained by MORE increase asymptotically to a limit, in line with their reward structure as described in Fig. 1. The much higher setting at which SFELLA's turning point

occurs (if it occurs at all) allows the agent to be more sensitive to gathering large rewards, which may be useful when the desired outcome is decreasing sensitivity to reward rather than downright reward indifference.

6.5 Safer AI for human values through MO decision-making

Overall, we found that SFELLA was able to exhibit one of our key design principles, loss aversion, at a similar level to TLO^A (Hypothesis 1) and MORE (Hypothesis 5). We partially confirmed Hypothesis 2, finding that SFELLA performed the BB task equally as well as TLO^A, while making fewer errors along the way. We confirmed Hypothesis 4, that performance appears to suffer as the algorithm becomes more granular, in a way mimicking the design of a thresholded agent. Hypothesis 6, that a transformation function could better meet the principle of balance between objectives by including a stronger reward sensitivity compared to MORE was confirmed.

How does SFELLA meet the design principles we proposed? SFELLA maintained a strong loss aversion, strongly preferring to avoid losses at a similar level compared to TLO^A and with only a small cost compared to MORE. It responded to changing positive reinforcement by paying a small price in other domains to greatly increase its reward received, in comparison to MORE, as described in Sect. 5.2.3. Though we haven't yet demonstrated SFELLA's ability to handle a large number of objectives, we have started that process by demonstrating its ability to balance three objectives. The agent has zero-point consistency, and in contrast to some 'low-impact AI' approaches, no distinction needs to be made between 'alignment' and 'performance' objectives, meaning that it could be suitable for balancing a variety of human objectives that span positive and negative domains.

Meeting the design principles in this way is a step forward for laying foundations for multi-objective modeling of human values by reducing risk through enabling them to be modeled in a way that avoids large losses in any particular objective. We hope this will enable future research to explore how human values could be balanced against one another by an artificial agent in a way that satisfies the human owners it is working for.

6.6 Future directions

Exploring conservative approaches to reinforcement learning and decision-making seems like a promising approach to advancing AI Safety, and multi-objective systems are one way forward.

6.6.1 Scaling calibration

In scaling calibration, a constant factor c_i is added to Eq. 2, scaling each objective value:

$$g(\vec{X}) = \sum_{i}^{n_{\text{obj}}} f(c_i X_i) \tag{9}$$

This step has not been implemented in this paper but we emphasize its possible use in the future.

When applying exponential transforms on each objective and then combining them in linear fashion, the scale of the operation is quite important. We designed SFELLA to primarily respond to z-scored input functions, i.e., most values typically appear between -3

and 3 (Fig. 1). 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 below 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' [29].

6.6.2 Wireheading

One possible failure mode for transformational AI systems has been described as 'wireheading', where 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 [7]. One solution to this involves ensuring that each proposed action is evaluated in terms of current objectives, so that changing the objectives themselves would not score highly on current objectives [8]. But a 'thin' conception of objectives, such as 'fulfill human preferences' might fail to sufficiently constrain the objective and leave too much of the function's implementation to re-learning and modification. 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. It may be that hardcoding more competing objectives which need all to be satisfied is a path to a safer AI less likely to wirehead its own systems.

6.6.3 Decision paralysis

We considered ways to implement maximin approaches such as that described by Vamplew et al. [34]. 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 or three 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' (or more precisely in this problem, avoiding conclusion of the task) evaluates more positively than any particular action. In fact, this may effectively be the outcome causing TLO^A's degraded performance in certain conditions. If TLO^A picks up two bottles, it is unable to let one go, but also unable to move to complete the task

In cases of decision paralysis, one way forward is for an agent to request clarification from a human overseer (see also Cohen and Hutter [6]). 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.

6.6.4 Optimal policies tend to seek power

So says Turner [30]. But what is the multi-objective response to this? If an agent has an incentive to maximize another agent's autonomy, or at least avoid minimizing it, then it will not tend to seek power beyond what is possible for maintaining that other agent's autonomy. We might design an effective reward system that avoids seeking power in order to maintain those items. This might be, for instance, an agent that deliberately preserves autonomy of another agent.

Human self-direction is among one of the universal human values [22, 23] and consequently should be one of the core objectives of human compatible AI.

6.7 Limitations

Some models of AI alignment [21] focus on 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. Where sufficient uncertainty exists, a sufficiently intelligent agent motivated to align to human preferences might respond by requesting clarification about the correct course of action from a human. This has similarities with the 'clarification request' under 'decision paralysis' described in this paper. But it remains to be seen whether a preference alignment approach can eliminate the need for explicit modeling of competing values. It might be that priming the agent to start with or to prefer certain shapes of utility functions might help in shaping its learning.

6.8 Conclusion

Continuous non-linear concave transformation functions could offer a way to find a compromise between multiple objectives where a specific threshold cannot be identified. This could be useful in situations where the trade-offs between objectives are not absolutely clear. We provide evidence that one such non-linear transformation function, SFELLA, is better able to respond to primary or alignment utility re-scaling.

Acknowledgements We wish to thank Peter Vamplew for his guidance on multi-objective utility functions, for providing the code we used to test our models, and for helpful comments as we edited the manuscript. 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. We are grateful to the AI Safety Camp for providing a forum for our team to meet and begin our project. Research reported in this publication was supported by the National Cancer Institute of the National Institutes of Health under Award Number 1R01CA240452-01A1, by the Emergent Ventures Program at the Mercatus Center at George Mason University under Award Number #3372, and by an EA Funds grant. RP and RK received payments for their work on this article. The content is solely the responsibility of the authors and does not necessarily represent the official views of any of the funders.

References

1. Armstrong, S., & Levinstein, B. (2017). Low impact artificial intelligences. arXiv:1705.10720 [cs] (May 2017).
2. Armstrong, S., & Mindermann, S. (2017). Impossibility of deducing preferences and rationality from human policy. CoRR abs/1712.05812 (2017). arXiv:1712.05812.
3. Bogosian, K. (2017). Implementation of moral uncertainty in intelligent machines. *Minds and Machines, 27*(4), 591–608. https://doi.org/10.1007/s11023-017-9448-z.
4. Bostrom, N. (2014). *Superintelligence*. Oxford University Press.
5. Byrnes, S. (2020). Conservatism in neocortex-like AGIs. https://www.alignmentforum.org/posts/c92YC89tznC7579Ej/conservatism-in-neocortex-like-agis.
6. Cohen, M.K., & Hutter, M. (2020). Pessimism about unknown unknowns inspires conservatism. In J. Abernethy, & S. Agarwal (Eds.), *Proceedings of thirty third conference on learning theory (Proceedings of machine learning research, vol. 125)*, PMLR, (pp. 1344–1373). http://proceedings.mlr.press/v125/cohen20a.html.
7. 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.
8. Dewey, D. (2011). Learning what to value. In *International conference on artificial general intelligence*. Springer, (pp. 309–314).
9. Gábor, Z., Kalmár, Z., & Szepesvári, C. (1998). Multi-criteria Reinforcement Learning. (pp.197–205).
10. Garrabrant, S. (2017). Goodhart taxonomy. https://www.alignmentforum.org/posts/EbFABnst8LsidYs5Y/goodhart-taxonomy.
11. Goodhart, C.A. (1984). Problems of monetary management: the UK experience. In *Monetary theory and practice*. (pp. 91–121). Springer.
12. Haidt, J. (2001). The emotional dog and its rational tail: a social intuitionist approach to moral judgment. *Psychological Review, 108*(4), 814.
13. Hayes, C.F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., et al. (2022). A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems, 36*(1), 26. https://doi.org/10.1007/s10458-022-09552-y.
14. Henrich, J., McElreath, R., Barr, A., Ensminger, J., Barrett, C., Bolyanatz, A., et al. (2006). Costly punishment across human societies. *Science, 312*(5781), 1767–1770.
15. Kahneman, D. (1979). Prospect theory: an analysis of decisions under risk. *Econometrica, 47*(1979), 278.
16. Peng, J., & Williams, R.J. (1996). Incremental multi-step Q-learning. *Machine Learning, 22*(1), 283–290. https://doi.org/10.1007/BF00114731.
17. Pratt, J.W. (1978). Risk aversion in the small and in the large. In *Uncertainty in economics*. (pp. 59–79). Elsevier.
18. Rawls, J. (2001). *Justice as fairness: a restatement*. Harvard University Press.
19. Roijers, D.M., & Whiteson, S. (2017). Multi-objective decision making. *Springer International Publishing*. https://doi.org/10.1007/978-3-031-01576-2.
20. Rolf, M. (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 tobotics (ICDL-EpiRob)*. (pp. 1–8). https://doi.org/10.1109/ICDL-EpiRob48136.2020.9278062. ISSN: 2161-9484.
21. Russell, S. (2019). *Human compatible: artificial intelligence and the problem of control*. Penguin.
22. Schwartz, S.H. (1992). Universals in the content and structure of values: theoretical advances and empirical tests in 20 countries. In *Advances in experimental social psychology*. (Vol. 25, 1–65). Elsevier.
23. Schwartz, S.H. (1994). Are there universal aspects in the structure and contents of human values? *Journal of Social Issues, 50*(4), 19–45.
24. Smith, B.J., & Read, S.J. (2022). Modeling incentive salience in Pavlovian learning more parsimoniously using a multiple attribute model. *Cognitive, Affective, & Behavioral Neuroscience, 22*(2), 244–257.
25. Sotala, K. (2016). Defining human values for value learners. In *AAAI workshop: AI, ethics, and society*.
26. Strathern, M. (1997). Improving ratings: audit in the British University system. *European Review, 5*(3), 305–321.
27. Sutton, R.S., & Barto, A.G. (2018). *Reinforcement learning: an introduction* second. The MIT Press.

28. Tom, S.M., Fox, C.R., Trepel, C., & Poldrack, R.A. (2007). The neural basis of loss aversion in decision-making under risk. *Science,315*(5811), 515–518. https://doi.org/10.1126/science.1134239. https://science.sciencemag.org/content/315/5811/515.full.pdf.
29. Turner, A.M., Hadfield-Menell, D., & Tadepalli, P. (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 .
30. Turner. A.M. (2019). Optimal farsighted agents tend to seek power. CoRR abs/1912.01683 (2019). arXiv:1912.01683.
31. Tversky, A., & Kahneman, D. (1991). Loss aversion in riskless choice: a reference-dependent model. *The Quarterly Journal of Economics, 106*(4), 1039–1061.
32. Vamplew, P., Smith, B.J., Kallstrom, J., Ramos, G., Radulescu, R., Roijers, D.M., Hayes, C.F., Heintz, F., Mannion, P., & Libin, P.J. K. et al. (2022). Scalar reward is not enough: a response to silver, Singh, Precup and Sutton. *Autonomous Agents and Multi-Agent Systems, 36*: 1–19.
33. Vamplew, P., Dazeley, R., & Foale, C. (2017). *Softmax Exploration Strategies for Multiobjective Reinforcement Learning, 263*(11), 74–86. https://doi.org/10.1016/j.neucom.2016.09.141.
34. Vamplew, P., Dazeley, R., Foale, C., Firmin, S., & Mummery, J. (2018). Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology, 20*(1), 27–40.
35. Vamplew, P., Foale, C., Dazeley, R., & Bignold, A. (2021). Potential-based multiobjective reinforcement learning approaches to low-impact agents for AI safety. *Engineering Applications of Artificial Intelligence, 100,* 104186. https://doi.org/10.1016/j.engappai.2021.104186.
36. Watkins, C.J.C.H., & Dayan, P. (1992). Q-learning. *Machine Learning, 8*(3), 279–292. https://doi.org/10.1007/BF00992698.