It's a matter of timescale: non-linear utility in successor features and multi-objective planning and learning *

Liam P.H. Mertens[1] , **Lucas N. Alegre**[2] , **Florent Delgrange**[1] ,
Diederik M. Roijers[1] , **Ann Nowé**[1] , **Peter Vamplew**[3]
[1]AI Lab, Vrije Universiteit Brussel, Belgium
[2]Institute of Informatics - Federal University of Rio Grande do Sul
[3]Federation University Australia
{liam.phi.h.mertens, florent.delgrange, diederik.roijers, ann.nowe}@vub.be,
lnalegre@inf.ufrgs.br, p.vamplew@federation.edu.au

Abstract

Time is of the essence when dealing with multiple reward signals and non-linear utility. In this paper we argue that the current main approaches in multi-objective RL (SER and ESR), and successor features, are insufficient. While each approach deals with non-linear effects on user utility on different timescales, none of them take into account that different effects happening on different timescales can happen within the same decision problem. We motivate that this can indeed be the case by an example, both intuitively and numerically, leading to a new perspective, and a significant and non-trivial gap in the literature.

1 Introduction

Planning and reinforcement learning agents, have been applied successfully in many domains. Many such agents rely on a scalar additive reward structure. This can be a highly limiting assumption in domains where distinct trade-offs between conflicting objectives need to be made. Instead, multi-objective planning and reinforcement learning methods consider multiple reward signals as a reward vector, with each component determining an objective that forms a part of a trade-off [Hayes *et al.*, 2022; Vamplew *et al.*, 2022b]. These reward vectors can then be accrued over time and then scalarised using a possibly non-linear utility function.

Multi-objective RL considers the utility of multiple accrued reward components over time. However, there are two ways of doing so: taking the expectation over accrued reward vectors (called the expected return or value) and considering the utility of this value, computed by applying a scalarisation function to the value vector. This is known as scalarised expected returns (SER) [White, 1982; Roijers *et al.*, 2013], and is perhaps the most prevalent way to consider utility. The underlying assumption here is that there is enough time for the expected returns to materialise, and that the utility is derived from this average. For example, consider the process mapping of a video editing application [Piscitelli, 2014]. A company that processes video editing jobs on their servers would processes large numbers of videos each day, and they would indeed be interested in the expected return of this process that consists of the average power consumption, average throughput, and average latency.

The second option is to accrue the rewards, apply the utility function on the resulting returns vector, and then take the expectation. This is known as expected scalarised returns (ESR) [Roijers *et al.*, 2018]. This would correspond to a scenario where the utility is derived from one trajectory. For example, imagine a patient that is undergoing treatment for an illness. That person will only undergo the treatment once, and therefore be interested in the expected utility of their own treatment outcome (not the utility of the average outcome) and thus will not have enough time to let the expected vector return materialise. For linear utility functions, ESR and SER are equivalent due to the distribution of the expectancy over the weighted sum. This is not the case under non-linear utility.

An alternative approach to multi-objective optimisation is the successor feature (SF) approach [Barreto *et al.*, 2017; Zhu *et al.*, 2024]. Unlike SER and ESR, which apply non-linear utility functions to vector returns only after they have been accrued over one or multiple trajectories, the SF approach evaluates non-linear utility at the level of individual states or transitions before any accumulation occurs, represented as a feature vector with components the non-linear successor features of these states/transitions. This corresponds to a scenario where utility effects happen on an immediate, per-step timescale. The feature vector is then linearly scalarised, which corresponds to planning and learning with a scalarised reward function in terms of multi-objective learning and planning. For example, imagine a car that hits something. The utility of that event may depend non-linearly on the force of impact, i.e., if you hit an obstacle very hard the risk of bodily injury increases non-linearly. These injury risks stack up over time, but it is not so that many subsequent low-force impacts accumulate to a single high-force one.

All three perspectives—SER, ESR, and the successor feature approach—make sense, and are motivated by different scenarios where the utility effects happen on different

*Presented at the Multi-Objective Decision Making workshop at IJCAI, 2026

timescales. As such, it seems like we have three sets of instruments to tackle problems that deal with non-linear effects on user utility, each on its own applicable timescale. There is however a key issue with this: the non-linear effects on user utility might be happening on different timescales *within the same decision problem*. To motivate this we take a (simplified) look at the domain of toxicity in Section 3, where toxic exposure can have adverse effects both immediately *and* on intermediate *and* on longer timescales. We work out a simplified numerical example to show that taking an SER, ESR, and SF approach lead to different policies with different outcome distributions. Furthermore, taking the perspective of adverse effects that happen on three different timescales simultaneously for this problem leads to different policies and different outcome distributions still.

In our opinion, this indicates that there is a clear gap in the multi-objective planning and reinforcement learning literature. So far we have not optimised for the effects on utility of rewards at different timescales simultaneously. In order to bridge this gap, we believe that we should redefine how we look at multi-objective problems, and develop new methodologies and algorithms to deal with this new perspective. In Section 6 we will discuss possible ways forwards.

2 Background

In RL, the framework for decision making is usually formalised as a *Markov decision process* (MDP). We care about scenarios where rewards are not necessarily scalar and where objectives may be competing and lead to trade-offs, giving rise to the notion of multi-objective optimisation in an MDP.

Formally, a *multi-objective Markov decision process* (MOMDP) is defined as a tuple $M = (\mathcal{S}, \mathcal{A}, p, \mathbf{r}, \mu, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p\left(\cdot \mid s, a\right) \in \Delta\left(\mathcal{S}\right)$ is the state transition function, $\mathbf{r}(s,a) \in \mathbb{R}^m$ is a multi-objective reward function with m objectives, μ is initial state distribution, and $\gamma \in [0,1)$ is a discount factor for future rewards.

A policy is a function mapping states to distributions over actions, $\pi \colon \mathcal{S} \to \Delta\left(\mathcal{A}\right)$, which intuitively prescribes which action an agent should select in each situation.[1]

We define the multi-objective return of a policy π as a random variable $\mathbf{G}^\pi := \sum_{t=0}^{\infty} \gamma^t \mathbf{R}_t$ from trajectories to vectors in $\mathbb{R}^m$, where the trajectories are drawn as $S_0 \sim \mu$, $A_t \sim \pi\left(\cdot \mid S_t\right)$, $\mathbf{R}_t = \mathbf{r}\left(S_t, A_t\right)$, and $S_{t+1} \sim p\left(\cdot \mid S_t, A_t\right)$ for all $t \geq 0$. The multi-objective value function of a policy π is the expected return, $\mathbf{v}^\pi(s) = \mathbb{E}_\pi[\mathbf{G}^\pi \mid S_0 = s]$.

2.1 Rewards in MORL

Knowing the multi-objective value functions of several policies is not enough, by itself, to decide which policy should be preferred. The value of a policy is a vector: one component may improve while another one gets worse. To turn such trade-offs into a decision, MORL therefore requires two ingredients: a *utility function* $u : \mathbb{R}^m \mapsto \mathbb{R}$, which assigns a scalar utility to a reward or return vector, and an *optimisation criterion*, which specifies when this utility is applied.

The placement of u matters. It determines whether we care about the utility of an average outcome, the average utility of whole trajectories, or the cumulative utility of each individual reward. These choices coincide for linear utility functions, but they can lead to different optimal policies when u is non-linear. Two optimisation settings have been widely used in the MORL literature: Scalarised Expected Returns, and Expected Scalarised Returns.

Scalarised Expected Returns

In the scalarised expected returns (SER) setting, the goal is to find

$$\pi^* \in \arg\max_\pi u(\mathbf{v}^\pi) = \arg\max_\pi u(\mathbb{E}_\pi[\mathbf{G}^\pi]), \tag{1}$$

given *any* utility function $u : \mathbb{R}^m \mapsto \mathbb{R}$. SER first averages the return vector of a policy and only then applies the utility function. Intuitively, it evaluates a policy by its long-run mean performance across objectives. This is natural when the same policy is used many times and the decision maker mainly cares about aggregate performance over a population of runs.

Expected Scalarised Returns

In the expected scalarised returns (ESR) setting, the goal is to find

$$\pi^* \in \arg\max_\pi \mathbb{E}_\pi\left[u\left(\mathbf{G}^\pi\right)\right], \tag{2}$$

given *any* utility function $u : \mathbb{R}^m \mapsto \mathbb{R}$. ESR applies the utility function to the complete return vector of each trajectory and only then averages across trajectories. It therefore captures how good an individual realised outcome is before taking expectations. This distinction is important when the utility function represents risk, fairness, saturation, or any other non-linear preference over complete outcomes.

Returns of Scalarised Rewards

A third possible setting is the approach used by default in single-objective RL: apply the utility function directly to the reward vector at each time step, and then maximise the expected return of the resulting scalar rewards. In returns of scalarised rewards (RSR), the scalar return is

$$G_u^\pi = \sum_{t=0}^{\infty} \gamma^t u(\mathbf{R}_t), \tag{3}$$

and the goal is to find

$$\pi^* \in \arg\max_\pi \mathbb{E}_\pi\left[G_u^\pi\right]. \tag{4}$$

RSR treats each time step as a separate trade-off. It is appropriate when the utility of an immediate reward vector is meaningful on its own, for example when one wants to penalise dangerous or undesirable instantaneous combinations of objectives, rather than only their cumulative effect over a full trajectory.

[1] For the different metrics defined in Sec. 2.1, we generally need both memory-based and stochastic policies. Memory can always be encoded into the state space. In terms of Pareto-optimality, deterministic policies may be used as the vertices of the related convex hull [Roijers *et al.*, 2013], but stochastic policies are required to fill the gaps on the curve (via a carefully designed mixture of deterministic ones [Vamplew *et al.*, 2009] or adding concave terms do the rewards [Lu *et al.*, 2022]).

Utility Functions
If u is linear (i.e., a simple weighted sum of the components of the reward or return vector), then the same policy will be optimal for SER, ESR, and RSR. However if a non-linear function is used for u (e.g. Chebyshev distance), then a different policy may be optimal for each of these settings. Importantly, this implies that we may have different optimal solution sets for each problem formulation [Hayes *et al.*, 2022].

2.2 Rewards in Succesor Features-based RL

The Successor Features (SFs) framework [Barreto *et al.*, 2017; Barreto *et al.*, 2020] was originally proposed to model multi-task RL problems, where each task is defined by a different reward function. It assumes that the reward function of an MDP is given by a linear function over a set of d *features* $\boldsymbol{\phi}(s,a) \in \mathbb{R}^d$:

$$r_{\mathbf{w}}(s,a) = \boldsymbol{\phi}(s,a) \cdot \mathbf{w}. \tag{5}$$

The SFs of a policy π represents the expected sum of the features: $\boldsymbol{\psi}^{\pi}(s,a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\phi}_t | S_0 = s, A_0 = a]$.

Intuitively, we can see this setting as an instantiation of the RSR setting where the vector reward $\mathbf{R}_t$ is replaced by the feature vector ϕ_t, and u is a linear function $u_{\mathbf{w}}(\phi_t) = \phi_t \cdot \mathbf{w}$. [Alegre *et al.*, 2022] showed that, in this case, SFs are equivalent to a multi-objective value function, and the SFs framework is equivalent to MORL under linear utility functions.

Beyond Linear Rewards in SFs
The expressiveness of SF-based approaches depends heavily on the definition of the features ϕ. Unlike the vector rewards in MORL that are typically specified by domain experts, the features ϕ can instead be learned by an agent. For example, these features can be learned via optimization of supervised criteria, e.g., to approximate a set of reward functions [Barreto *et al.*, 2018], or via unsupervised objectives, e.g., to approximate the Laplacian of the MDP [Chandrasekar and Machado, 2025] or to identify clusters of the state space [Bagot *et al.*, 2025].

Importantly, if the feature vector is a one-hot encoding indicating the current state of the agent in the state space, $\phi(s) = [\mathbb{1}_{S_1}(s), \cdots, \mathbb{1}_{S_d}(s)]^{\top}$ where $d = |\mathcal{S}|$, then any reward function, r, can be represented as a vector $\mathbf{w} \in \mathbb{R}^{|\mathcal{S}|}$, where $\mathbf{w} = [r(s_1) \ ... \ r(s_{|\mathcal{S}|})]^{\top}$. This implies that non-linear functions of the reward vector (Section 2.1) would be unnecessary. The Forward Backward (FB) representation [Touati and Ollivier, 2021; Touati *et al.*, 2022] fits in this context by learning a high-dimensional representation that, under some dimensionality constraints/assumptions, can express the optimal policy for any possible reward function.

We observe a tendency for SF-based approaches to increase their expressiveness (the family of problems they can solve) by discovering more powerful feature representations, while maintaining a linear utility function.

In contrast, in the MORL literature, most works typically consider a low-dimensional reward vector, but expand the family of problems that MORL techniques can solve by employing non-linear utility functions. We believe that, although orthogonal, it makes sense to combine both directions when tackling some real-world problems.

That being said, the multi-objective RSR criterion with non-linear u, can be implemented in successor features by simply appending the existing features ϕ_t by a non-linear combination of these features: $\phi_t \leftarrow \phi_t \cup \{\tilde{u}(\phi_t)\}$.[2] This feature then gets accrued over time as the utility, and a weight of 1 is placed on this new feature. SER and ESR in general however, cannot be implemented using traditional successor feature approaches, as after accruing rewards over time, only linear combinations are allowed.

3 Motivating Example: Toxicity

As mentioned above, the optimisation criteria (SER, ESR, and RSR) say something about the impact of—reward or feature—signals on user utility. This happens at different timescales: across multiple episodes for SER, over one episode for ESR, and at a single-timestep for successor features/RSR. All of these make sense in one setting or another. However, we would like to argue here, that they can also make sense all at once. It is just a matter of timescale.

Let us turn to the example of exposure to substances and their toxicity. Toxicity effects typically follow a non-linear dose-response curve, and thus leads to non-linear utility. A high dosage in a short time can lead to severe immediate adverse effects. This is known as *acute exposure* (one dose, or accumulating exposure up to 24 hours), and could correspond to a timestep-based application of a non-linear (utility function) before the rewards/penalties are added up at all.

However, just because a substance does not hurt you when you are exposed to it at a lower dosage once, does not mean you can keep getting exposed to it. In fact, in toxicology the following exposure time-lengths are often distinguished *sub-acute* (24 hours to 28 days), *sub-chronic* (less than 90 days), and *chronic* exposure [Denny and Stewart, 2024].

Let's consider a small company with three certified people to work with toxic substances. These workers have different experience levels, and therefore also different risk exposure risks for different tasks. Let's imagine we aim to optimise the monthly work schedule, with an employee taking the same work station for a day, doing one task. For simplicity lets imagine there are 3 tasks in total that are the same each day. Let's take the dose each of the three employees get as three separate reward functions. What happens when we take different types of exposure into account?

3.1 Intuition

First let us see what would happen if we would take different types of toxic exposure into account in an intuitive manner, before looking at a numerical example.

Chronic exposure. For chronic exposure, we would consider the long-term damage of being exposed to a toxin in small amounts over a long period of time. This would mean we ought to look at the exposure over the course of multiple months or even years. In RL terms this would correspond

[2] This assumes all features in ϕ_t are reward components. If this is not the case we would instead have to apply a composite function $u \circ \tilde{r}$, where $\tilde{r}$ is the translation of the successor features to the rewards in each objective.

to multi-objective RL (MORL) under the scalarised expected returns (SER) criterion [Hayes *et al.*, 2022]. Furthermore, because the response to toxins are typically sigmoid shaped—meaning that the toxic effects start increasing slowly for low dosages and then increase ever faster before flattening off again—this means we probably just have to spread out the average toxic exposure over all employees as much as possible so that each individual's long-term average dose is low, leading to the least averse effects.

Sub-acute exposure. For sub-acute exposure, we would be interested in the total dosage an employee would get over the entire month. This would require us look at the possible outcomes for each month, as a probability distribution, and look at the expected adverse effects of the cumulative dosages for the entire month (again by applying a probably sigmoidal response curve). It would probably also be important to track closely the dosage the employee already received up until a given day, and dynamically adapt our schedule to that. In RL terms this would correspond to expected scalarised returns (ESR) [Roijers *et al.*, 2018].

Acute exposure. For acute exposure, we would be interested in the immediate effects of a high dose within 24 hours. This would entail considering the dosage an employee gets as a result of a task, and then applying the response curve directly to that. Taking the average of that would correspond to the average risk of acute exposure on a daily basis. We could model this using successor features, by directly considering the toxic response—per timestep—as a state feature, corresponding to RSR.

So what would a responsible employer have to do now? In the decision theoretic literature we seem to have the choice between SER, ESR, or RSR as an optimisation criterion. Each of these would imply only one of the different timescales for toxic exposure. But that does not suffice; a responsible employer should have their eye on *every* type of toxic exposure, not just one. We should optimise for all at once.

We argue that this indicates a critical gap in the literature, i.e., different problems may require us to look at (cumulative) rewards on different timescales, but importantly *different timescales may have to be taken into account at the same time*.

3.2 Numerical example

Now we turn our example into a numerical decision problem, and will show that different policies are optimal for different settings. The decision problem consists of 3 employees of varying skill levels and 3 tasks with varying toxicity (table 1). The goal is to assign the employees optimally over time given a possibly non-linear utility function that limits the maximal toxic dosage per employee either per day, per month, over multiple months or over all possible time frames simultaneously. This problem can be modeled as a MOMDP with state space $\mathcal{S}$ defined as all permutations of employees, with their positions indicating their assigned task (e.g. [emp1, emp2, emp3] indicates that employee 1 is assigned to task 1, employee 2 to task 2 and employee 3 to task 3) and action space $\mathcal{A}$ as the finite discrete set of permutation indices such that $|\mathcal{A}| = 3!$. Note that the state needs to be augmented with the accrued reward in order to accommodate non-linear utility under ESR and SER [Vamplew *et al.*, 2024; Reymond *et al.*, 2023; Roijers *et al.*, 2018]. The vectorial reward function is defined as $\mathbf{r}(s_t, a_t) = [-r_1(s_t, a_t), -r_2(s_t, a_t), -r_3(s_t, a_t)]$, where r_i indicates the toxic dose that each employee is exposed to on a given day. These doses are defined as $r_i = \max(0, (1-S_i)\cdot T_j + \varepsilon)$, with T_j being the toxicity of the task currently assigned to employee i and S_i the corresponding skill level of that employee. The doses are slightly stochastic, subject to the noise $\varepsilon \sim \mathcal{N}(0, 10)$. We note that in a successor feature formulation, the received dosages would be appended to the state, rather than given as a reward vector. We consider episodes of 30 days long to simulate work schedules over a month.

Table 1: The environment parameters. Each employee initially starts at their respective task ($s_0 = [0, 1, 2]$).

	Skill level
Employee 1	0.75
Employee 2	0.60
Employee 3	0.2

	Toxicity
Task 1	70
Task 2	25
Task 3	10

In order to model and mitigate the three types of toxic exposure highlighted above, we propose three utility functions.

The acute utility function determines the limit on acute exposure and as such, should be applied to the reward at each timestep:

$$u_{acute}(\mathbf{r}) = \sum_{i=1}^{3} \frac{1}{1 + e^{-.5(\mathbf{r}_i + 25)}} \tag{6}$$

The episodic utility function considers sub-acute and chronic exposure (e.g. over one or multiple episodes) and thus should be applied over the returns or values:

$$u_{episodic}(\mathbf{G}_t) = \sum_{i=1}^{3} \frac{1}{1 + e^{-.02(\mathbf{G}_{t,i} + 400)}} \tag{7}$$

Note that in the case of SER optimization $u_{episodic}(\mathbf{V}^{\pi})$ should be computed instead of over the return $\mathbf{G}_t$ as under ESR.

The combined utility function should be a combination of both u_{acute} and $u_{episodic}$ (applied to both the returns and the value). This is a novel approach to solving multi-objective sequential decision-making problems using a utility-based approach.

4 Experiments

To showcase the need for a unified view we performed experiments on the example highlighted in section 3.2 for the three optimisation criteria: ESR, SER and RSR. Finally, we combined the three criteria through a multi-objective natural evolution strategy (MONES) [Glasmachers *et al.*, 2010; Salimans *et al.*, 2017]. Note that the experiments serve as an illustrative proof of concept to show distinct differences between optimisation criteria.

4.1 Algorithms

In order to show the differences in policies learned under each optimization criterion, we used algorithms that are tailored for those settings [3].

EUPG (ESR) Expected Utility Policy Gradient (EUPG) [Roijers *et al.*, 2018] is a multi-objective adaptation of the REINFORCE algorithm [Williams, 1992]. It is designed to learn optimal policies under ESR for non-linear utilities and achieves this by incorporating the accrued reward during learning and applying the utility over the sum of the accrued return and future returns.

NLPPO (SER) Optimising for non-linear utility under the SER criterion requires computing the value function first before computing the utility function. This is achieved by modifying an offline variant of proximal policy optimisation (PPO) [Schulman *et al.*, 2017; Meng *et al.*, 2023] to apply the utility over the critic network's output, as proposed by [Röpke *et al.*, 2025].

SFDQN (SF/RSR) We simulated a successor feature (/RSR) approach for this problem by transforming the problem into a linear multi-objective RL problem through the application of u_{acute} (Eq. 6), which is a non-linear transformation with a subsequent linear combination using uniform weights (all employees counting equally) (as in Eq. 5). This non-linear transformation of the reward function corresponds to the learning of successor features that are non-linear w.r.t. the states in the problem, given that the original state space were extended with a non-linear utility applied to the reward components. After application of the linear scalarisation on the non-linear features, we are left with a single-objective RL problem that can be solved using deep Q networks (DQN) [Mnih *et al.*, 2013].

MONES (combined perspective) Optimising for the combined utility over different timesteps requires a novel approach, as directly combining either of the methods above is infeasible because scalarisation happens over different time windows and does not distribute over time due to the non-linearity of the utilities. We propose to use a natural evolution strategy to learn the search distribution over policy parameters that optimises the combined utility per timestep, per episode and on average over multiple episodes. This combination is achieved by multiplying all three utilities to get the fitness function of the parameter population, inspired by the AND operator in fuzzy logic. In order to ensure equal contribution of each utility to the total, we divide u_{acute} by 30, the episode length, to get the average acute utility. This leads to equal contribution of each sub-utility to the total due to the normalisation effect of the sigmoidal utility functions.

4.2 Setup

The hyperparameters for the algorithms are given in table 2. Each experiment was performed using 3 different seeds on an Apple M5 CPU.

Table 2: Hyperparameter configurations for the evaluated algorithms.

Hyperparameter	Value
Shared	
Discount factor (γ)	1
Episode length	30
Optimizer	Adam
EUPG (ESR)	
Learning rate	1e-4
Replay buffer size	1e5
Learning steps	3e6
NLPPO (SER)	
Actor learning rate	2.5e-4
Critic learning rate	2.5e-4
Clip range (ϵ)	0.2
Entropy coefficient	0.05
GAE λ	0.95
Learning steps	3e6
SFDQN (RSR)	
Learning rate	1e-4
Replay buffer size	1e5
Target net update freq.	1000
Exploration fraction	0.5
Learning steps	2e6
MONES (Combined)	
Population size	50
Eval episodes per member	60
Learning rate	1e-4
Noise std. dev. (σ)	0.1
Learning steps (generations)	5e6

4.3 Results

When comparing the return distributions of each method in Fig. 1, we could notice differences between assignments. Optimising under ESR led to a switching strategy being learned that kept the toxicity penalty nicely around the threshold of 400 for all employees. Note that this implies exceeding the daily intake limit at times. Looking at the optimal policy under SER, the return distribution has less kurtosis. This is the result of the policy preferring to keep employees 1 and 2 under the threshold by rotating them, but in turn 'sacrificing' the least experienced employee 3.[4] By maximising the utility for 2 employees instead of nicely balancing, it aims to get a higher total utility. Note however, that the expected return under SER lies close to the expected return under ESR for employee 1 and 2, although with increased variance. This shows why SER optimisation is not optimal in cases where risk-awareness is necessary. The need for sacrificing employee 3 stems from the rollouts where employee 1 and 2 are kept well under the toxicity threshold for the episode, leading to

[3] The code for our experiments is available at https://github.com/liammertens/MORL-timescale/tree/main

[4] Please note that this is a numerical example problem, and we would not recommend this in practice.

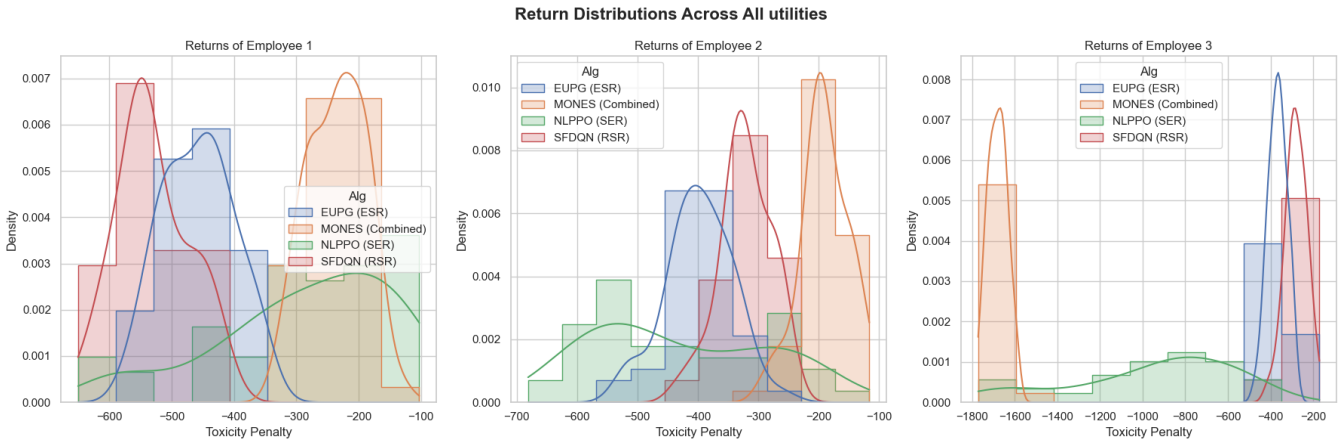

Figure 1: Comparison of the return distributions resulting over 50 policy rollouts.

optimal *expected* utility. Optimisation under RSR leads to a switch-averse policy that prefers to keep employees working on the tasks that correspond to their skill level, as this strategy is certain to keep working and no additional requirements on total toxicity are given. Finally, the combined method learns to shuffle, but mainly prefers to put employee 3 on the most difficult task, even more than under SER, leading to optimisation for the utilities of the two most experienced employees. One explanation for this phenomenon could be that the combination of all 3 utilities allows the agent to maximise one or two of them and neglect the other, leading to the 'sacrifice' scenario. E.g. the agent is not willing to risk exceeding the daily thresholds for employee 1 and 2 in exchange for the total utility of employee 3, so they are kept on the less intensive tasks.

The sacrificing policies being optimal for SER and the combined objective are a clear indication that the problem is too hard, as not all employees can be safe from chronic exposure. Especially in the combined objective, it appears that the safety of the third employee is given up, indicating a very clear safety issue for the employer. However, if you look at the RSR or ESR optimisation, the switching policies do seem to balance the toxicity well, and the employees are safe. As such, there seems to be a lot of value in looking at the different timescales at once, if we want to guarantee the safety of all. Clearly the employer needs more employees, or better hazard mitigation measure, to lower the exposure risks when looking at the combined objective, while they might not think so looking solely at ESR (or RSR). In this—admittedly highly simplified/abstract—example, looking at different timescales at once is thus truly *essential*.

To conclude we observe that there are problems for which looking at different timescales individually leads to very different results than looking at the combined utility effects on different timescales simultaneously. For this we used three standard algorithms for RSR, ESR and SER, and a heuristic approach (MONES) for the combined utility of different timescales. We believe that these results call for further investigation into, and specialised algorithms for, combined timescale utility optimisation.

5 Related Work

The majority of MORL algorithms consider only a single optimisation criteria. For example, [Cai *et al.*, 2023] shows how to, given an MOMDP under ESR, construct a MDP with augmented state space and scalar reward function that induces the same optimal policy as the original MOMDP. [Vamplew *et al.*, 2022a] discussed a scenario where the optimal policy might maximise SER subject to a satisfactorily small level of variation in returns between episodes (reduced-variance SER), but their proposed algorithm is restricted to the specific case of finding SER-optimal deterministic policies for environments with stochastic rewards but deterministic state dynamics. More recently [Röpke *et al.*, 2023] introduced the distributional undominated set (DUS), which in theory contains all optimal policies under both SER and ESR criteria.

At a theoretical level, studies have examined how the choice of utility function or optimisation criterion impacts on the policies that are available to a RL or MORL agent. [Subramani *et al.*, 2024] proposes a hierarchy of RL formalisms, and shows that, in theory, the SER (Outer MORL) formalism can express all policy orderings that ESR (Inner MORL) can express. [Rodriguez-Soto *et al.*, 2024] shows a characterisation of the utility functions for which an associated optimal policy exists, and (2) a characterisation of the types of preferences that can be expressed as utility functions.

In the literature on multi-task RL and successor measures/features, a few works have explored similar problems. Regarding approaches that learn distributional value functions, [Wiltzer *et al.*, 2024] introduces a distributional analogue to the SFs. In particular, they show that it allows agents to perform zero-shot *risk-sensitive* policy evaluation, which can be seen as a type of non-linear utility. The Forward-Backward (FB) representation [Touati and Ollivier, 2021; Touati *et al.*, 2022] is another relevant framework that allows to approximate (under some dimensionality constraints) the optimal policy for any scalar reward function, fitting the formalism presented in Section 2.1. Recently, [Bagatella *et al.*, 2026] studies how to optimize for general non-linear utility functions of the successor measure of a policy. We believe this type of method is a good example of combining expres-

sive reward representations with non-linear utilities.

Another relevant line of work that extends the expressivity of SF-based methods is the Option Keyboard (OK) [Barreto *et al.*, 2019; Barreto *et al.*, 2020]. Instead of following a policy specialized to a fixed preference vector $\mathbf{w}$, an OK agent is capable of dynamically changing this vector (and the associated policy being followed) at each time step, similar to dynamic preferences in MORL [Abels *et al.*, 2019]. This type of technique has been shown to be capable of optimally solving some classes of non-linear reward functions [Alegre *et al.*, 2025].

6 Discussion

In this paper, we have shown the need for a utility-based perspective in multi-objective planning and reinforcement learning that works on multiple timescales at once. We have shown this intuitively and then numerically on a toxicity example.

In our example—for reasons of simplicity—we left out the sub-chronic toxicity that is also known from the toxicology literature. In terms of timescale, this would fall between SER and ESR; it goes beyond one execution of the policy (encompassing about a month), but would not be quite the average longtime outcome either. There is some discussion over the exact timescale of what it means to be sub-chronic[5], but if we take the 90 days mark it would correspond to roughly three policy executions in our example. We could look at the distribution of the summed doses in three policy executions, which would be a smoother distribution of values. This could be added as a fourth timescale.

Although we only used one abstract example in this paper, we do believe that these types of problems are actually quite common. Consider for example companies that are interested in the average time a customer's order needs to be completed, in order to optimise serving costs. However, they might also be interested in the individual customer's waiting times at the same time, as if a handful of customers might have to wait very long compared to the main bulk, this can lead to very bad reviews and lead to damage to the company's reputation. This would suggest at least a combined SER-ESR perspective and corresponds to an average reward RL problem [Mahadevan, 1996]. In the medical domain (possibly AI-assisted) radiation therapy [Van der Veen *et al.*, 2019] is spread out over multiple sessions to reduce the radiation damage to surrounding organs, while making sure the accumulated dose over time is enough to kill a tumor. This suggests a combined RSR-ESR perspective. So indeed, we suspect that a lot of critical real-world decision problems exhibit utility properties on multiple timescales.

With the numerical example, we have shown that different policies work better for different timescales and the corresponding optimisation criteria. We argued that we need to develop comprehensive methodology and algorithmic approaches to deal with different timescales simultaneously, as policies that work well for SER, ESR, or RSR may not suffice for a combination of the three.

We believe that a key aspect of any solution that works on multiple timescales simultaneously will be a distributional one, i.e., to learn a distributional return and distributional reward function. However, we do note that even if we have those, policy optimisation is far from trivial. A change in policy can have effects on multiple timescales, leading to a highly complex utility landscape. The AI community has developed multiple ways to tackle such complex utility landscapes though, including policy gradients, evolutionary strategies, etc. So we are hopeful that our community can and will find good approaches for this issue.

[5] See e.g., https://tinyurl.com/subchronic.

Ethical Statement

The toxic exposure example in this paper is used purely for illustrative purposes. We are not advocating for organisations to manage employees' exposure to toxins in this manner.

More broadly, we believe that the greater expressivity provided by the use of multiple rewards and optimisation criteria enhances the AI agents' capacity to include ethical considerations explicitly during decision-making [Vamplew *et al.*, 2018].

Acknowledgments

We acknowledge the following funding: LPHM was supported by the VUB; FD was supported by the Belgian Flemish AI Research Program and the "DESCARTES" iBOF project; DMR was supported by the PEER project, which has received funding from the EU's Horizon Europe Research and Innovation Programme, under Grant Agreement number #101120406. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This work was supported by Kunumi Institute. The authors thank the institution for its financial support and commitment to advancing scientific research.

We would like to thank Siri Willems (of imec AI labs) for her input and feedback on the medical examples.

References

[Abels *et al.*, 2019] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 11–20. PMLR, May 2019.

[Alegre *et al.*, 2022] Lucas Nunes Alegre, Ana L. C. Bazzan, and Bruno C. Da Silva. Optimistic Linear Support and Successor Features as a Basis for Optimal Policy Transfer. In *Proceedings of the 39th International Conference on Machine Learning*, pages 394–413. PMLR, June 2022.

[Alegre *et al.*, 2025] Lucas N. Alegre, Ana L. C. Bazzan, Andre Barreto, and Bruno Castro da Silva. Constructing an Optimal Behavior Basis for the Option Keyboard. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025.

[Bagatella *et al.*, 2026] Marco Bagatella, Thomas Rupf, Georg Martius, and Andreas Krause. Soft Forward-Backward Representations for Zero-shot Reinforcement Learning with General Utilities, February 2026. arXiv:2602.06769 [cs].

[Bagot *et al.*, 2025] Louis Bagot, Lucas N. Alegre, Steven Latre, Kevin Mets, and Bruno Castro da Silva. Successor clusters: A behavior basis for unsupervised zero-shot reinforcement learning. *Transactions on Machine Learning Research*, 2025.

[Barreto *et al.*, 2017] Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, Long Beach, CA, USA, 2017. Curran Associates, Inc.

[Barreto *et al.*, 2018] Andre Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Zidek, and Remi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 501–510, Stockholm, Sweden, July 2018.

[Barreto *et al.*, 2019] Andre Barreto, Diana Borsa, Shaobo Hou, Gheorghe Comanici, Eser Aygün, Philippe Hamel, Daniel Toyama, Jonathan hunt, Shibl Mourad, David Silver, and Doina Precup. The Option Keyboard: Combining Skills in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Barreto *et al.*, 2020] André Barreto, Shaobo Hou, Diana Borsa, David Silver, David Silver, and Doina Precup. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences of the United States of America*, 117(48):30079–30087, 2020.

[Cai *et al.*, 2023] Xin-Qiang Cai, Pushi Zhang, Li Zhao, Jiang Bian, Masashi Sugiyama, and Ashley J. Llorens. Distributional pareto-optimal multi-objective reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 15593–15613, Red Hook, NY, USA, December 2023. Curran Associates Inc.

[Chandrasekar and Machado, 2025] Siddarth Chandrasekar and Marlos C Machado. Towards An Option Basis To Optimize All Rewards. 2025.

[Denny and Stewart, 2024] Kevin H. Denny and C.W. Stewart. Chapter 6 - acute, subacute, subchronic, and chronic general toxicity testing for preclinical drug development. In Ali S. Faqi, editor, *A Comprehensive Guide to Toxicology in Nonclinical Drug Development (Third Edition)*, pages 149–171. Academic Press, third edition edition, 2024.

[Glasmachers *et al.*, 2010] Tobias Glasmachers, Tom Schaul, and Jürgen Schmidhuber. A natural evolution strategy for multi-objective optimization. In *Parallel Problem Solving from Nature (PPSN XI), Part I*, pages 627–636, 09 2010.

[Hayes *et al.*, 2022] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning: Cf hayes et al. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[Lu *et al.*, 2022] Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-Objective Reinforcement Learning: Convexity, Stationarity and Pareto Optimality. In *The Eleventh International Conference on Learning Representations*, sep 2022.

[Mahadevan, 1996] Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22(1):159–195, March 1996.

[Meng *et al.*, 2023] Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. Off-Policy Proximal Policy Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9162–9170, June 2023.

[Mnih *et al.*, 2013] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[Piscitelli, 2014] Roberta Piscitelli. *Pruning techniques for multi-objective system-level design space exploration*. PhD thesis, Universiteit van Amsterdam, 2014.

[Reymond *et al.*, 2023] Mathieu Reymond, Conor F. Hayes, Denis Steckelmacher, Diederik M. Roijers, and Ann Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23, April 2023.

[Rodriguez-Soto *et al.*, 2024] Manel Rodriguez-Soto, Juan A. Rodriguez-Aguilar, and Maite Lopez-Sanchez. An analytical study of utility functions in multi-objective reinforcement learning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pages 77726–77747, Red Hook, NY, USA, December 2024. Curran Associates Inc.

[Roijers *et al.*, 2013] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.*, 48:67–113, 2013.

[Roijers *et al.*, 2018] Diederik M Roijers, Denis Steckelmacher, and Ann Nowé. Multi-objective reinforcement learning for the expected utility of the return. In *Proceedings of the Adaptive and Learning Agents workshop at FAIM*, volume 2018, 2018.

[Röpke *et al.*, 2023] Willem Röpke, Conor F. Hayes, Patrick Mannion, Enda Howley, Ann Nowé, and Diederik M.

Roijers. Distributional multi-objective decision making. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5711–5719. International Joint Conferences on Artificial Intelligence Organization, August 2023.

[Röpke *et al.*, 2025] Willem Röpke, Mathieu Reymond, Patrick Mannion, Diederik M Roijers, Ann Nowé, and Roxana Rădulescu. Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, pages 1774–1783, Richland, SC, June 2025. International Foundation for Autonomous Agents and Multiagent Systems.

[Salimans *et al.*, 2017] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[Subramani *et al.*, 2024] Rohan Subramani, Marcus Williams, Max Heitmann, Halfdan Holm, Charlie Griffin, and Joar Skalse. On The Expressivity of Objective-Specification Formalisms in Reinforcement Learning, February 2024.

[Touati and Ollivier, 2021] Ahmed Touati and Yann Ollivier. Learning One Representation to Optimize All Rewards. In *Advances in Neural Information Processing Systems*, volume 34, pages 13–23. Curran Associates, Inc., 2021.

[Touati *et al.*, 2022] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does Zero-Shot Reinforcement Learning Exist? In *The Eleventh International Conference on Learning Representations*, September 2022.

[Vamplew *et al.*, 2009] Peter Vamplew, Richard Dazeley, Ewan Barker, and Andrei V. Kelarev. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In Ann E. Nicholson and Xiaodong Li, editors, *AI 2009: Advances in Artificial Intelligence, 22nd Australasian Joint Conference, Melbourne, Australia, December 1-4, 2009. Proceedings*, Lecture Notes in Computer Science, pages 340–349. Springer, 2009.

[Vamplew *et al.*, 2018] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and information technology*, 20(1):27–40, 2018.

[Vamplew *et al.*, 2022a] Peter Vamplew, Cameron Foale, and Richard Dazeley. The impact of environmental stochasticity on value-based multiobjective reinforcement learning. *Neural Computing and Applications*, 34(3):1783–1799, 2022.

[Vamplew *et al.*, 2022b] Peter Vamplew, Benjamin J Smith, Johan Källström, Gabriel Ramos, Roxana Rădulescu, Diederik M Roijers, Conor F Hayes, Fredrik Heintz, Patrick Mannion, Pieter JK Libin, et al. Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, 36(2):41, 2022.

[Vamplew *et al.*, 2024] Peter Vamplew, Cameron Foale, Conor F. Hayes, Patrick Mannion, Enda Howley, Richard Dazeley, Scott Johnson, Johan Källström, Gabriel Ramos, Roxana Rădulescu, Willem Röpke, and Diederik M. Roijers. Utility-based reinforcement learning: Unifying single-objective and multi-objective reinforcement learning, 2024.

[Van der Veen *et al.*, 2019] J Van der Veen, S Willems, S Deschuymer, D Robben, W Crijns, F Maes, and S Nuyts. Benefits of deep learning for delineation of organs at risk in head and neck cancer. *Radiotherapy and Oncology*, 138:68–74, 2019.

[White, 1982] DJ677750 White. Multi-objective infinite-horizon discounted markov decision processes. *Journal of mathematical analysis and applications*, 89(2):639–647, 1982.

[Williams, 1992] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992.

[Wiltzer *et al.*, 2024] Harley Wiltzer, Jesse Farebrother, Arthur Gretton, Yunhao Tang, André Barreto, Will Dabney, Marc G. Bellemare, and Mark Rowland. A distributional analogue to the successor representation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 52994–53016, Vienna, Austria, July 2024. JMLR.org.

[Zhu *et al.*, 2024] Chuning Zhu, Xinqi Wang, Tyler Han, Simon Shaolei Du, and Abhishek Gupta. Distributional Successor Features Enable Zero-Shot Policy Optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.