Safety embedding: designing safe environments with lexicographic deep reinforcement learning

Arnau Mayoral-Macau [a,*], Manel Rodriguez-Soto[a], Daniele Meli[b], Martí Sánchez-Fibla[a] and Juan Antonio Rodríguez-Aguilar[a]

[a]IIIA-CSIC, Spain
[b]Department of Computer Science, University of Verona, Verona, Italy

Abstract. Safety in reinforcement learning (RL) is crucial for agents in both physical and digital settings, requiring them to complete tasks while behaving safely. This work-in-progress report proposes a method for designing environments that enforce safe behaviour in RL agents. Building on multi-objective RL, our approach embeds an environment with two objectives (task and safety) into one where safety is always prioritised. Unlike prior embedding methods with high computational costs, we introduce a simplified algorithm that enables faster design of safe environments, extending applicability to larger and more complex domains. We also demonstrate its effectiveness on a continuous control task, a setting previously unexplored by embedding techniques.

1 Introduction

Autonomous agents are becoming increasingly present in both real and digital worlds. Consequently, their irruption in different fields and their performance on a wide range of tasks are raising concerns about whether these agents are truly reliable, trustworthy, and safe [8, 7, 2]. Such concerns are especially pertinent to decision-making agents, whose decisions may imply several risks regarding, for instance, safety or ethics. Within decision-making algorithms, reinforcement learning (RL) [22] is being widely used to train agents for real-world tasks, thanks to its ability to produce adaptable and high-performing behaviours from experience. RL algorithms that leverage neural networks (deep reinforcement learning, or DRL) have demonstrated strong results in generalising complex tasks in real-world settings [14, 5]. However, neural networks are opaque, which ultimately leads to black-box behaviours that introduce additional safety concerns. Literature on safe RL [4] tries to mitigate the risks of DRL, applying so-called cost functions. A cost function measures undesirable behaviours or safety constraint violations that an agent should avoid. The agents that minimise a cost function below a certain threshold are therefore agents with a safe behaviour. Employing cost functions shifts the problem from a single-objective to a multi-objective perspective, where both the reward and (one or more) cost functions have to be regarded. By establishing a threshold on the cost functions, the problem can be framed as a constrained optimisation problem [1, 3], where only policies that satisfy the constraints are considered feasible policies. Another option is to treat each cost function as an objective that must be *lexicographically* prioritised over the primary objective of the agent [24, 27, 23]. These two approaches have different characteristics. Whilst constraining the learning with specific thresholds offers more flexibility when a certain amount of cost is feasible, it also requires more complex algorithms and prior knowledge of the cost function's range [21]. On the other hand, lexicographic optimisation is stricter on the prioritisation of the objectives, allowing for less flexibility, but generally comes with a smaller associated computational cost [24].

Constrained RL has been widely used for safety in realistic environments such as robotics or autonomous driving [9, 11], including scenarios with continuous action spaces. However, little work has been done with lexicographic algorithms, despite lexicographic prioritisation being perfectly suited to the idea of "*safety over performance*". When examining safe continuous control problems, the situation is even worse, as, to the best of our knowledge, no lexicographic algorithm has been tested for continuous action spaces. One explanation is the focus on value-based algorithms for solving lexicographic problems, and the difficulties those algorithms encounter with continuous action spaces. Within the family of policy-gradient algorithms, only [21] and [27] have approached lexicographic algorithms, both introducing lexicographic versions of PPO [20] and both calling it *Lexicographic PPO* (LPPO). On one hand, [27]'s implementation is incompatible with continuous control, as it uses a dynamic action masking module specifically suited for discrete action spaces. On the other hand, whilst [21]'s LPPO is compatible with continuous action spaces, their experiments concentrate only on discrete scenarios, as they demonstrate local and global convergence for their algorithms on discrete problems.

In the context of ethical decision-making, recent literature [15, 12] has shown that it is possible to embed multiple objectives (e.g., ethical and individual task objectives) into a single scalarised reward function, starting from a lexicographic (hence strict) priority ordering. These algorithms allow environment designers to combine distinct objectives into a single one, where the importance of the objectives is the environment designer's choice. In the case of ethics, the environment designers can impose ethics as the most important objective. Environment design is a compelling solution to value alignment because it allows a designer to enforce ethical learning.

With a single objective, a learning agent cannot disregard ethics, as it cannot manipulate or prioritise the embedded objectives differently. Conversely, if the reward signals are given separately, the environment designer loses control over how agents prioritise the distinct reward signals. However, existing embedding approaches are time-consuming because they require the construction of a partial Convex Hull [15, 16, 12], a process that involves computing optimal policies

* Corresponding Author. Email: arnau.mayoral@iiia.csic.es

for various scalarised environments. When these optimal policies are difficult to learn due to environment complexity, the embedding techniques become even more expensive to compute, limiting their applicability.

Although these environment-centred techniques have only been used to align agents with ethics, their lexicographic nature makes them suitable for safety applications. Therefore, analogously to the way in which they prioritise ethics, we can prioritise safety and develop techniques to design safe environments.

Against this background, we present a work-in-progress *Safety Embedding* algorithm that simplifies the existing *Approximate Embedding*[12] algorithm, making it computationally less demanding. Our work-in-progress contributes to the safety RL literature in the following key aspects:

- We explore the applicability of lexicographic learning algorithms, specifically LPPO [21], in continuous action spaces. This includes evaluating its performance in the MetaDrive environment [10], where an autonomous car agent must simultaneously control both acceleration and steering through continuous actions. To the best of our knowledge, this is the first application of DRL lexicographic techniques to continuous actions and partial observability.
- We show that the internal coefficients LPPO [21] uses to scalarise loss functions can be repurposed to scalarise the reward functions of a multi-objective environment in order to create a single-objective safe environment.
- We perform a preliminary assessment of the generalisation capabilities of the scalarised reward function computed through the embedding, to investigate if the same embedding can be used across multiple scenarios. That is, in the case of Metadrive, check if the scalarisation function computed for a certain driving situation is able to incentive safe behaviour for other unseen situations.

2 Background

Multi-objective sequential decision-making can be framed as a multi-objective Partially Observable Markov Decision Process (MOPOMDP) [17, 18], where an agent acts in an environment, altering its state, receiving multiple rewards, and perceiving only partial observations. Formally:

Definition 1. *A Multi-Objective Partially Observable Markov Decision Problem is defined by a tuple* $\mathcal{M} = \langle S, \mathcal{A}, R^{1,\dots,m}, T, O, \mathcal{O}, \gamma \rangle$ *where* S *is the state space of the environment,* $\mathcal{A}$ *is the action set of the agent,* $R : S \times A \to \mathbb{R}^m$ *are the reward functions,* $T : S \times A \times S \to [0, 1]$ *is the transition function of the environment,* O *is a finite set of observations and the function* $\mathcal{O} : A \times S \times O \to [0, 1]$ *represents the probabilities over the agent's possible observations* o *given the state* s *and action* a*. Finally,* $\gamma \in [0, 1)$ *is the discount factor, indicating future rewards' importance are on the current state.*

On a MOPOMDP, a stochastic policy $\pi(a|h_t)$ assigns the probability of selecting action a given the past history $h_t = (o_0, a_0, o_1, a_1, \dots, o_t, a_t)$ up to the current time step t. Throughout a complete history h (until a terminal state), a policy accumulates multiple rewards and produces a vector representing the achievement of the multiple objectives. The expected discounted return vector $\vec{V}^{\pi}(s) = \mathbb{E}_{h\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t \vec{R}(s_t, a_t)\right]$, namely the *value vector* represents the expected overall performance of the policy π over all the objectives and possible histories h.

These value vectors $\vec{V}^{\pi}(s)$ can be combined using a weight vector $\vec{w} \in \mathbb{R}^m$ to produce a single scalar objective. However, linear prioritisations are not desirable to represent non-linear preferences among objectives. In domains like safety, where there is a non-linear prioritisation in favour of safety, significant domain knowledge or extensive experimentation may be required to determine a weight vector that avoids dangerous trade-offs between safety and performance in other tasks [6]. Non-linear prioritisations are then preferable, in the form of a *lexicographic order* $\ell = \{V_1 \succeq V_2\}$, where objective V_1 is always prioritised over V_2. This leads to defining *Lexicographic POMDP*[24] as a variant of MOPOMDPs:

Definition 2. *A Lexicographic Partially Observable Markov Decision Process (LPOMDP) is a tuple* $\mathcal{M} = \langle S, \mathcal{A}, R^{1,\dots,m}, T, O, \mathcal{O}, \ell, \gamma \rangle$*, where* ℓ *is the lexicographic order among the objectives, that is, any possible permutation of* $\{1, \dots, m\}$ *where the first objective in the order is the higher-ranked objective. The rest of the elements of the tuple are defined exactly like in a MOPOMDP.*

2.1 Lexicographic DRL

A common state-of-the-art DRL technique is to train an agent with two different neural networks. These actor-critic algorithms have an actor neural network that models the policy $\pi(a|h_t, \theta)$, while a critic neural network is trained to predict the value of a certain observation $V(o|\phi)$ given its parameters ϕ. The critic network is trained by observing the returns the actor is obtaining throughout the learning, and the actor is updated by increasing the probability of the actions with better estimations from the critic.

In multi-objective DRL, each reward function R^j has its own loss function $\hat{K}^j(\theta)$, defined as the average loss of the current training batch, which generates gradients aimed at maximising the expected discounted return in the j-th objective. Lexicographic DRL algorithms [21] aim to minimise each of the loss functions $\hat{K}^j(\theta)$ without increasing the loss on higher-ranked loss functions $\hat{K}^k(\theta)$ in the lexicographic order ($k < j$). To transform the constrained problem into an easier unconstrained optimisation, a common practice is to use Lagrangian relaxation [3, 28], where the constraints are added to the primary objective as penalties weighted by Lagrange multipliers λ's.

For lexicographic actor-critic methods [21], the Lagrange multiplier λ, together with the lexicographic order ℓ, is used to scalarise the distinct loss functions into a single loss signal. Thus, without entering into much detail, $c_t^j(\lambda_t^j, \ell)$ are the scalarisation coefficients that will be used to train the actor at time-step t. For the sake of simplifying notation, we will just write c_t^j. Thus, the scalarised loss function can be expressed as follows:

$$\hat{K}(\theta) := \sum_{j=1}^{m} c_t^j \hat{K}^j(\theta) \tag{1}$$

During training, the coefficients λ_t^j should increase when the current loss on objective j, $\hat{K}^j(\theta)$, exceeds the average loss $\hat{k}_j$ for the same objective, computed over a buffer of b_s past losses, and decrease otherwise. This update rule can be relaxed by including a tolerance τ, $\hat{K}^j(\theta) > \hat{k}_j + \tau$, which should start at a higher value at the beginning of training and gradually decrease to zero. The inclusion of this tolerance parameter ensures adequate exploration of the state–action space and prevents early convergence to overly conservative policies.

With the correct adjustment of each λ^j and, therefore, the c_t^j coefficients, formulating lexicographic optimisation problems becomes simple and at a low computational cost. In practice, this approach can be integrated with any actor-critic algorithm (i.e. PPO, A2C, TRPO), and training it with regular temporal difference methods [20, 19].

3 Formalising the safety embedding problem

We aim to design a single-objective environment in which an agent learns to perform its primary task while exhibiting safe behaviour. To achieve the learning of safe policies, we must create an environment where safe behaviour is optimal. In the literature [15, 16, 12], this type of problem has been addressed only in the context of ethics. To design ethical environments, first, they encode the primary task and the ethics alignment into different reward functions. Then, both objectives are combined into a single objective in which ethics has a lexicographic preference over performance on the primary task. Since safety and ethics share a major importance over the primary task, we argue that the same approach can be applied to safety.

Consider an original *source* LPOMDP, where ℓ establishes a preference for safety objectives. Consequently, the optimal policy π_r for such an LPOMDP corresponds to safe behaviour that maximises the primary task. We refer to such policies as safe policies. We then aim to design a *target* environment with a single reward function such that π_r is also optimal for this target environment.

Using a linear scalarisation function $f(\vec{w}_s) = \vec{w}_s \cdot \vec{R}$ to combine the objectives, the safety embedding problem is to find a weight vector $\vec{w}_s$ that is sufficiently large to ensure that safe behaviour is optimal over any unsafe behaviour. Formally:

Problem 1. *Let $\mathcal{M} = \langle S, \mathcal{A}, R^{1,\dots,m}, T, O, \mathcal{O}, \ell, \gamma \rangle$ be a LPOMDP where ℓ establishes a prioritisation in favour of safety. Then,* ***the safety embedding problem*** *is that of finding a weight vector $\vec{w}_s$ that can define a POMDP $\mathcal{M}' = \langle S, \mathcal{A}, R = \vec{w}_s \cdot \vec{R}, T, O, \mathcal{O}, \gamma \rangle$ such that any lexicographically dominant policy in $\mathcal{M}$ attains higher scalarised expected return in $\mathcal{M}'$ than any dominated policy.*

When the safety embedding problem is solved, we can build the target POMDP by utilising the scalarised function $R = f(\vec{w}_s)$, which will lead learning agents to safe behaviour.

The next section proposes a new method for solving the safety embedding problem and relates it with the state of the art.

4 Safety embedding process

Although no existing embedding algorithm addresses safe environment design, related literature tackles similar embedding problems. [15, 16] introduced *optimal embedding* (OE), the first techniques for designing ethical environments. Optimal embedding used learning algorithms with convergence guarantees to find an optimal environment design. Recent work [12] introduced an approximate embedding (AE) method using DRL to address scalability in ethical multi-agent design with large state spaces and partial observability. AE employs lexicographic DRL to compute a reference policy and leverages PPO to train agents in scalarised POMDPs, using binary search to identify the weight required for a policy equivalent to the reference, enabling scenarios previously infeasible with tabular methods.

Overall, the existing methods share a key limitation: they require the computation of a Convex Hull. While OE costs may be justified by optimal solutions, AE remains computationally expensive for an approximate approach, with its binary search comprising 85.71% of total costs in original experiments. This stems from the cost of POMDP learning, often needing several attempts in complex settings. Thus, we argue that minimising learning processes is crucial for reducing embedding technique costs.

In contrast to these methods that construct an aligned environment as a result of exploring different scalarisation weights, there are algorithms that directly learn an aligned policy. Lexicographic algorithms such as LPPO [21] do so by *dynamically* changing the importance of each objective using Lagrange multipliers during learning. Thus, throughout training, the impact of each objective on the overall policy learning changes to adapt to the lexicographic order. That is, when a more prioritised objective is losing performance, it is given more weight. In the case of LPPO, this continual adaptation is achieved through a linear scalarisation of the loss functions (Eq. 1) using a vector of coefficients $\vec{c}_t$ at each time step t.

Since the LPPO algorithm optimises a single scalarised loss signal derived from a multi-objective loss, we say that LPPO addresses a problem similar to the safety embedding problem. Arguably, as LPPO learns aligned policies, this suggests that during learning, the scalarisation via the coefficients $\vec{c}_t$, and the relative importance assigned to each objective, has incentivised aligned behaviour.

We then propose the *Safety Embedding* (SE) algorithm. The algorithm takes one source LPOMDP $\mathcal{M}$, with a specific lexicographic prioritisation ℓ, as input, and returns a POMDP $\mathcal{M}'$ where the maximisation of a single reward signal leads the agents to a behaviour that abides by the prioritisation ℓ. The algorithm follows three steps:

1. Compute a reference policy π_r on an LPOMDP using LPPO and tracking its internal coefficients $\vec{c}_t$.
2. Select the coefficients $\vec{c}_T$, where T is the final time-step, as the scalarisation weight vector $\vec{w}_s = \vec{c}_T$.
3. Build and return a POMDP $\mathcal{M}'$ using the weights $\vec{w}_s$ to scalarise the reward function as:

$$R = \sum_{j=1}^{m} w_s^j \cdot R^j \tag{2}$$

This algorithm is general for embedding a lexicographic order ℓ in a single reward signal. Therefore, it can be used for any domain where lexicographic prioritisation is appropriate, including ethical embedding. However, in this work-in-progress, we focus on safety.

5 Experimental Analysis

With our experiments, we want to highlight several aspects of the Safety Embedding algorithm. At a high level, our primary objective is to show that SE can compute a weight vector $\vec{w}_s$ that creates a POMDP where the optimal policy is to be safe. As a first exploration of SE, we will test it for an initial multi-objective setting with an individual task and a *single* safety objective ($m = 2$). Crucially, we target continuous action spaces, an unexplored area in both embedding algorithms and lexicographic DRL literature.

On a technical level, we investigate the robustness of SE in identifying scalarisation weights $\vec{w}_s$, particularly whether it converges to the same weights across runs. We further hypothesise that, while not guaranteed, the learned weights transfer across similar environments, making the embedding generalisable and extending the algorithm's applicability. For example, weights learned in a smaller environment could be reused in a larger one with more obstacles, where computing the embedding would be more expensive.

The following list details how we empirically investigate the above-mentioned ideas:

1. **Robustness.** We run LPPO multiple times with different seeds to prove that different learning instances converge to similar scalarisation weights, with low standard deviation, thus demonstrating the robustness of the Lagrangian method in finding a specific weight vector that incentivises safe behaviour.
2. **Embedding accuracy.** When an agent is trained in a single-objective environment designed with a reward function such as Eq. 2 with the corresponding scalarisation weights, the agent

learns a safe policy similar to the reference policy in terms of environment-specific safety metrics.
3. **Generalisation.** We assess generalisation by setting an agent to learn in an environment using the scalarised reward function computed for a different environment.

All the experiments will be conducted in MetaDrive [10], a customisable autonomous driving environment. The next section specifies the MetaDrive environment we use: we use the default observations, but we set a reward function that better suits our goal.

5.1 The MetaDrive environment

In MetaDrive [10], the agent's objective is to reach the end of the road as quickly as possible while navigating a variety of obstacles, including traffic, damaged vehicles, fences, and cones. To enable a controlled evaluation of the SE algorithm, we eliminated all stochastic elements, rendering the environment fully deterministic. Future work will explore the performance of SE under stochastic conditions. We model the MetaDrive environment as follows:

Observation The partial observations of the agent include: (1) a 240-dimensional vector lidar that detects cars and cones with a maximum detection distance of 70 meters, (2) a 40-dimensional vector lidar that detects the sides of the road, (3) an ego state vector that includes the current steering, heading, velocity and relative distances to the left and right boundaries and, (4) navigation information that guides the vehicle towards the goal through a set of checkpoints. All this forms a 296 vector of continuous values.

Figure 1. Experimental environments with right and left turns and obstacles to force lane change.

Action space We configured MetaDrive in its continuous control task, where the action of the agent is a vector of two real numbers $a^t, a^s \in [-1., 1.]$, representing the throttle (from maximum braking force to maximum acceleration) and steering (from maximum steering to the left to maximum steering to the right).

Reward function We define two distinct reward functions. The **individual objective** R^p uses the following reward:

1. A speed component received each step that rewards going as fast as possible: $\frac{v}{v_{max}} \cdot c_1$, where v is the current speed of the car, v_{max} is the maximum speed of the car, and c_1 is a scalar coefficient that modulates the importance of speed on the overall reward function. In our configuration of the environment, $v_{max} = 80$, $c_1 = 0.2$.
2. A cutting distance component received each step that rewards how close the car is to the next checkpoint in comparison to the previous step: $c_2 \cdot \Delta distance$, where $\Delta distance$ is the distance reduced (in environments' map coordinate units) to the checkpoint w.r.t the last step, and c_2 is a scalar coefficient that modulates the importance of this component. We set c_2 to 0.3.
3. When reaching the destination, the agent receives reward:

$$300 \cdot \left(1 - 0.9 \cdot \frac{t}{T_{\text{max}}}\right),$$

where t is the current time step and T_{max} is the maximum number of steps allowed in an episode.

In contrast, the **safety objective** R^s is a sum of multiple components received at each step t depending on the state s:

1. Staying centered within the current lane $r_c(s_t) \in [0, 0.125]$ where a perfectly aligned car receives a reward of 0.125.
2. Driving in the rightmost lane $r_r(s_t) \in [0, 0.5]$ where driving in the centre of the right lane yields a reward of 0.5.
3. Maintaining a minimum speed of 20 km/h $r_{ms}(s_t) \in [-0.7, 0]$ where a stopped car receives a penalty of -0.7 and a car above the minimum speed does not get penalised.

At each step, we add these three components and give the outcome to the agent. Collisions or leaving the road terminate the episode with a -2000 penalty. The multi-level definition of R^s has a hierarchy: r_c, r_r and r_{ms} that encourages safe driving but may be violated in justified cases, such as lane changes. From a constraint optimisation view, staying right, centred, and above minimal speed are soft constraints [13], which may be surpassed to avoid breaching hard constraints like crashes or leaving the road.

5.2 LPPO Robustness

We trained an LPPO agent on the track in Fig. 1 (left) for 45M steps and with 3 different seeds. Fig. 2 shows the averaged learning curves for the different seeds in terms of accumulated return. Notably, the learning of both objectives is stable, with low variance, which smoothly leads the agents to convergence. Importantly, it is even more stable for the (prioritised) safety objective.

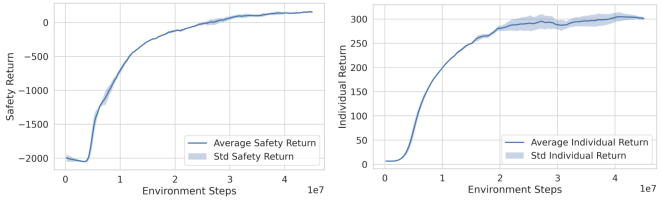

Figure 2. Learning curves for both objectives during LPPO training.

Metric	Reference Policy π_r	Scalarised Policy π_s	Generalised Policy π_g
Avg Speed (km/h)	20.82 ± 0.239	7.53 ± 0.06	33.43 ± 2.44
Crash-free episodes	$98.7\% \pm 1.5\%$	$100\% \pm 0\%$	$98\% \pm 1.6\%$
Reached goal	$98.7\% \pm 1.5\%$	$0\% \pm 0\%$	$98\% \pm 1.6\%$
Nº Time-steps to Goal	998 ± 26	1100 ± 0	651 ± 49
Avg. Normalised r_r	0.44 ± 0.019	0.83 ± 0.05	0.0 ± 0.0
Avg. Normalised r_c	0.7 ± 0.019	0.915 ± 0.017	0.65 ± 0.01

Table 1. Averaged metrics (over the three seeds) of policy rollouts. r_r: normalised reward per step for driving in the rightmost lane; r_c: normalised reward per step for driving in the centre of the current lane.

The first column in Table 1 shows the performance of the three LPPO policies computed on three different seeds, averaged over 100 episodes. We observe that the agent learns a safe policy without crashing in 98.7% of the episodes. Regarding the individual objective, the reference policy follows a speed marginally above the minimum speed required, and reaches the goal on average at $t = 998$, almost at the end of the episode (which ends at $T_{max} = 1100$).

To proceed with the SE algorithm, we must retrieve the value of the scalarisation coefficient c_t^s LPPO uses at the end of the training.

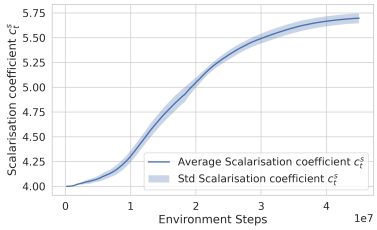

Figure 3. Average and standard deviation of the scalarisation coefficient c_t^s throughout the training on three different seeds.

Fig. 3 shows how the coefficient c_t^s changes during the learning of three runs of LPPO. We can see how c_t^s increases while training and by the end it slowly converges to $c_T^s = 5.7$ on average and with a very low standard deviation. These results indicate that LPPO is robust in finding a weight large enough to produce safe policies.

Since, by default, we set the less prioritised objective, in this case, the individual objective, to have a fixed coefficient of $c_T^p = 1$. There fore the weight vector $\vec{w}_s = (c_T^s, c_T^p) = (1, 5.7)$ would then be used to build the target $\mathcal{M}'$ POMDP.

5.3 *Embedding accuracy*

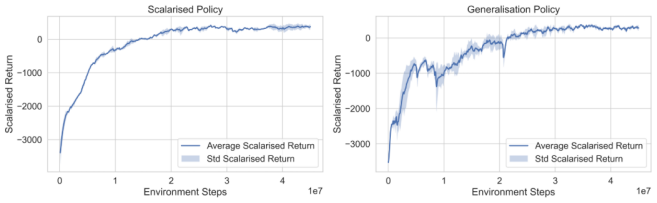

Figure 4. Training curves of the scalarised policy π_s (left) and generalisation policy π_g (right) averaged for 3 different seeds each.

With the computed weight vector $\vec{w}_s$, we define a scalarised POMDP suitable for standard single-objective RL and train a policy using Proximal Policy Optimisation (PPO) [20], denoted as the scalarised policy (π_s). Comparing π_s with π_r assesses the effectiveness of the embedding.

The learning curve of π_s (Fig. 4, left) shows stable training, with all seeds converging to similar scalarised returns. Performance, averaged over 100 evaluation episodes and three seeds (Table 1, second column), reveals that the agent adopts an overly cautious strategy: it moves at 5.8 km/h, well below the 20 km/h minimum speed, and thus fails to reach the second obstacle or the goal within the 1100-step limit (Fig. 1, left). While this prevents direct comparison with the reference policy, π_s exhibits strong safety near the first obstacle, with zero collisions and high scores for the soft safety metrics r_r and r_c, which benefit from slower speeds.

5.4 *Generalisation*

We apply the scalarised reward function from Section 5.2 to a new environment with a more complex road (Fig. 1, right), which adds a lane and obstacle compared to training. Using this fixed reward, we trained an agent and obtained the generalisation policy π_g, with learning curves shown in Fig. 4. Training with three seeds produced similar outcomes, though with some instability. Direct comparison to prior policies is not possible due to track differences, but analysis of π_g reveals key findings: it matches the reference policy in crash-free rate despite higher complexity, reaches the goal in 712 steps with an average speed of 30 km/h, and while maintaining lane-centring, fails to stay in the rightmost lane.

6 Discussion

6.1 *Current results*

Our preliminary experiments suggest that both the scalarised and generalisation policies have converged to local optima. The scalarised policy adopted a conservative strategy, characterised by slow, risk-averse behaviour, while the generalisation policy disregarded secondary objectives that had minor presence on the rewards, such as adhering to the right lane. This is a consequence of having all safety considerations collapsed into a single reward; lower-valued components might be under-represented. To address the early convergence to conservative policies, we plan on increasing the minimum speed penalty component r_{ms}. With the current value, an agent is able to accumulate a safety reward each step while being below the threshold. For instance, when driving at 10 km/h, the agent receives a penalty of -0.35, which can be worth it since driving centred on the right lane has a maximum reward of $r_c + r_r = +0.625$, which is easier to attain by driving slower.

Since all policies have a high percentage of safe runs and are at least as good as the reference policy in that regard, we can consider that SE has effectively built a safe environment.

6.2 *Comparison with approximate embedding*

The Safety Embedding algorithm presents a promising and computationally efficient alternative to the approximate embedding algorithm. By bypassing the costly binary search inherent to AE, a process that requires solving multiple scalarised learning problems, SE achieves a substantial improvement in scalability.

Another distinction is that AE approximates the minimal safety weight (w_s), while SE's weight is not necessarily close to this minimum. However, SE's w_s can serve as an upper bound for AE's search, narrowing the interval and speeding convergence. Thus, combining both methods may benefit weight minimisation. In conclusion, SE is a notable advancement, as it directly learns a scalarisation weight without exhaustive search.

In conclusion, the SE approach represents a notable advancement for embedding techniques, as it directly learns a scalarisation weight, rather than relying on exhaustive trial-and-error searches.

7 Conclusions and Future Work

Safety Embedding offers a scalable solution to the challenges of existing multi-objective embedding methods, making the design of aligned environments feasible in large-scale and continuous domains. While future work will extend experiments to validate our hypotheses (see Section 6), several directions stand out. First, SE can support environments with multiple alignment reward functions, as its Lagrangian foundation is effective in multi-constraint settings [26, 25]. This could help distinguish between hard and soft safety constraints, simplifying reward design. Second, LPPO could be refined to minimise the Lagrangian multiplier, thereby reducing the scalarisation coefficient and keeping the final weight w_s close to its minimum effective value. Finally, extending SE to multi-agent domains [12] by combining it with PPO variants such as ILPPO or MALPPO could enable the creation of complex, aligned multi-agent environments.

References

[1] E. Altman. *Constrained Markov Decision Processes*. PhD Thesis, INRIA, 1995. URL https://inria.hal.science/inria-00074109/document.

[2] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[3] J. Dai, J. Ji, L. Yang, Q. Zheng, and G. Pan. Augmented proximal policy optimization for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7288–7295, 2023. URL https://ojs.aaai.org/index.php/AAAI/article/view/25888. Issue: 6.

[4] J. Garcıa and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015. URL https://www.jmlr.org/papers/volume16/garcia15a/garcia15a.pdf.

[5] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

[6] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), Apr. 2022.

[7] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.

[8] J. Hernández-Orallo, F. Martínez-Plumed, S. Avin, and S. O. Heigeartaigh. Surveying Safety-relevant AI characteristics. In *AAAI workshop on artificial intelligence safety (SafeAI 2019)*, pages 1–9. CEUR Workshop Proceedings, 2019. URL https://riunet.upv.es/handle/10251/146561.

[9] F. Khan, W. Feng, Z. Wang, T. Huang, X. Liu, Y. Cui, and W. Weijun. Safe Reinforcement Learning for Vision-Based Robotic Manipulation in Human-Centered Environments, June 2025. URL https://www.researchsquare.com/article/rs-6736564/v1. ISSN: 2693-5015.

[10] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou. MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning, July 2022. URL http://arxiv.org/abs/2109.12674. arXiv:2109.12674 [cs].

[11] K. Lin, Y. Li, S. Chen, D. Li, and X. Wu. Motion Planner With Fixed-Horizon Constrained Reinforcement Learning for Complex Autonomous Driving Scenarios. *IEEE Transactions on Intelligent Vehicles*, 9(1):1577–1588, Jan. 2024. ISSN 2379-8904. doi: 10.1109/TIV.2023.3273857. URL https://ieeexplore.ieee.org/document/10120952.

[12] A. Mayoral-Macau, M. Rodriguez-Soto, E. Marchesini, M. López-Sánchez, M. Sanchez-Fibla, A. Farinelli, and J. A. R. Aguilar. Designing ethical environments using multi-agent reinforcement learning. In *The Seventeenth Workshop on Adaptive and Learning Agents*. URL https://openreview.net/forum?id=uJxmULkvJE.

[13] P. Meseguer, F. Rossi, and T. Schiex. Chapter 9 - Soft Constraints. In F. Rossi, P. van Beek, and T. Walsh, editors, *Foundations of Artificial Intelligence*, volume 2 of *Handbook of Constraint Programming*, pages 281–328. Elsevier, Jan. 2006. doi: 10.1016/S1574-6526(06)80013-1. URL https://www.sciencedirect.com/science/article/pii/S1574652606800131.

[14] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang. Solving Rubik's Cube with a Robot Hand, Oct. 2019. URL http://arxiv.org/abs/1910.07113. arXiv:1910.07113 [cs].

[15] M. Rodriguez-Soto, M. Lopez-Sanchez, and J. A. Rodriguez-Aguilar. Multi-objective reinforcement learning for designing ethical environments. In *IJCAI*, pages 545–551, 2021.

[16] M. Rodriguez-Soto, M. Lopez-Sanchez, and J. A. Rodriguez-Aguilar. Multi-objective reinforcement learning for designing ethical multi-agent environments. *Neural Computing and Applications*, pages 1–26, 2023.

[17] D. M. Roijers. *Multi-objective decision-theoretic planning*. s.n.], [S.l., 2016. OCLC: 1151484905.

[18] R. Rădulescu, P. Mannion, D. M. Roijers, and A. Nowé. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, 34(1), Apr. 2020. ISSN 1387-2532, 1573-7454. doi: 10.1007/s10458-019-09433-x. URL http://link.springer.com/10.1007/s10458-019-09433-x.

[19] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, Aug. 2017.

[21] J. Skalse, L. Hammond, C. Griffin, and A. Abate. Lexicographic Multi-Objective Reinforcement Learning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3430–3436, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/476. URL https://www.ijcai.org/proceedings/2022/476.

[22] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[23] A. Tercan and V. S. Prabhu. Thresholded Lexicographic Ordered Multiobjective Reinforcement Learning, Sept. 2024. URL http://arxiv.org/abs/2408.13493. arXiv:2408.13493 [cs].

[24] K. Wray, S. Zilberstein, and A.-I. Mouaddib. Multi-objective mdps with conditional lexicographic reward preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[25] T. Xu, Y. Liang, and G. Lan. CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11480–11491. PMLR, July 2021. URL https://proceedings.mlr.press/v139/xu21a.html. ISSN: 2640-3498.

[26] Y. Yao, Z. Liu, Z. Cen, P. Huang, T. Zhang, W. Yu, and D. Zhao. Gradient shaping for multi-constraint safe reinforcement learning. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, pages 25–39. PMLR, June 2024. URL https://proceedings.mlr.press/v242/yao24a.html. ISSN: 2640-3498.

[27] H. Zhang, Y. Lin, S. Han, and K. Lv. Lexicographic Actor-Critic Deep Reinforcement Learning for Urban Autonomous Driving. *IEEE Transactions on Vehicular Technology*, 72(4):4308–4319, Apr. 2023. ISSN 1939-9359. doi: 10.1109/TVT.2022.3226579. URL https://ieeexplore.ieee.org/abstract/document/9969953. Conference Name: IEEE Transactions on Vehicular Technology.

[28] Y. Zhang, Q. Vuong, and K. Ross. First Order Constrained Optimization in Policy Space. In *Advances in Neural Information Processing Systems*, volume 33, pages 15338–15349. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/af5d5ef24881f3c3049a7b9bfe74d58b-Abstract.html.