Generating Explainable Counterfactual Policies through Temporal Logic Queries

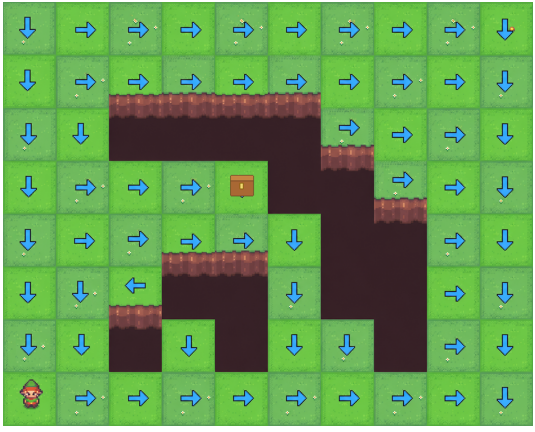
Arnaud Lequen Clément Legrand-Lixon Léo Saulières

Linköping University Univ. Lille (CRIStAL) Univ. Toulouse (INRAE-MIAT)

Outline

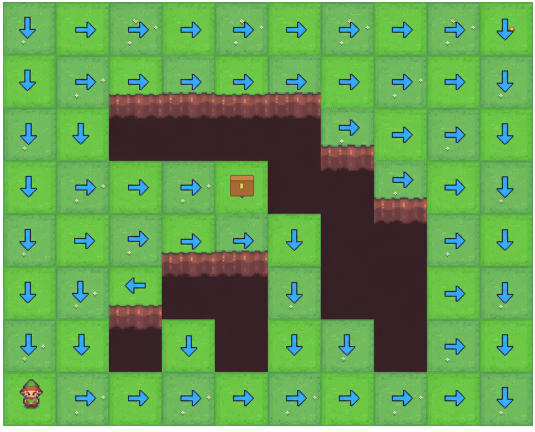
Running example: Cliffwalking

Main objective: Reach the bottom-right cell through a shortest path



Running example: Cliffwalking

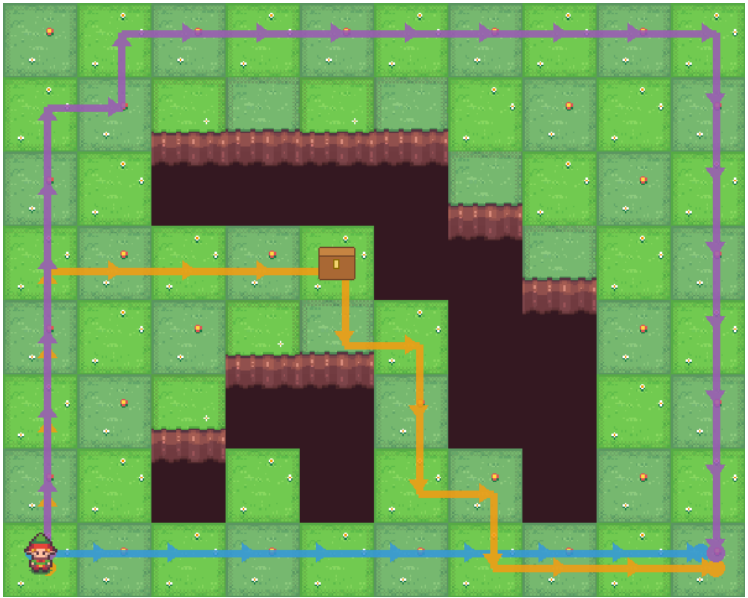
Main objective: Reach the bottom-right cell through a shortest path



What if we *also* wanted to...

- pass by the treasure?
- avoid being near cliffs?

Running example: Cliffwalking

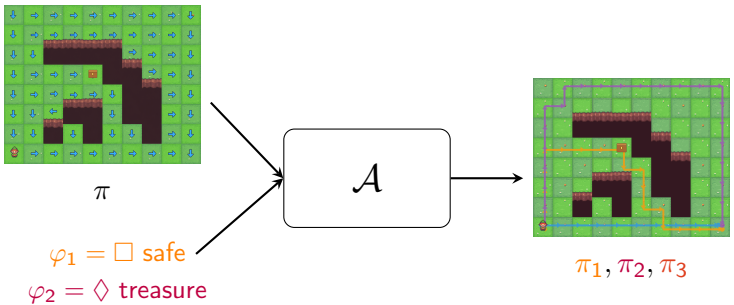


Altering Policies

Context

- Learning a policy from scratch is costly
- Sometimes, the user had in mind a slightly different policy

How to minimally deviate from a policy to also achieve a desirable property?



Goal: let the user explore a set of **counterfactual policies** obtained by expressing preferences about π's behavior.

Key ingredients

Multi-objective Reinforcement Learning

- **Multi-policy** algorithms generate a whole Pareto front
  - Shows compromises between objectives
  - Resilient to bad specifications from the user
  - All policies learned in one go

Linear Time Logic

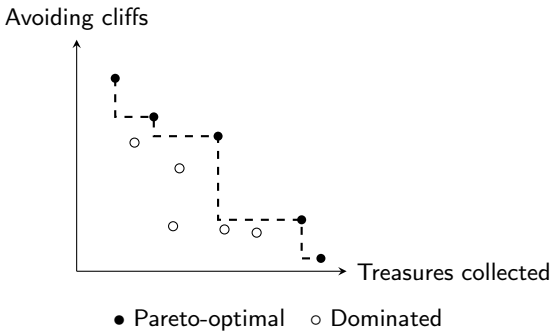
- More intuitive to express than reward functions
- Express temporally-extended goals
- Expose the structure of the reward to the learning algorithm

(Multi-Objective) Markov Decision Processes

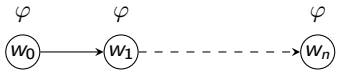
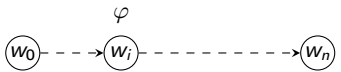
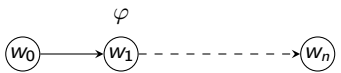
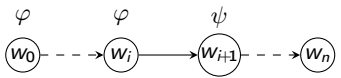
An MOMDP is an MDP with a reward signal R in $\mathbb{R}^d$.

- The MDP describes the environment: (properties of) states, actions, etc.
- R gives one reward per objective.

Solutions are policies which are on the **Pareto front**:



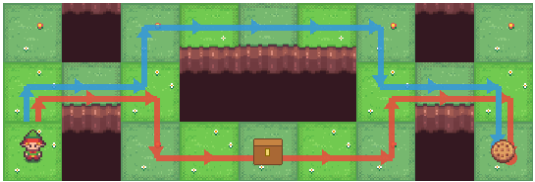
LTL$_f$ at a glance

$\Box\varphi$	Always	
$\Diamond\varphi$	Eventually	
$\bigcirc\varphi$	Next	
$\varphi\,\mathcal{U}\,\psi$	Until	

Examples

- $\Diamond$`treasure` – the agent eventually grabs the treasure.
- $\Box$`safe` – the agent always stays on safe cells.
- $\Box(\neg$`safe` $\Rightarrow \bigcirc$`safe`$)$ – never two unsafe steps in a row.

Counterfactual policies



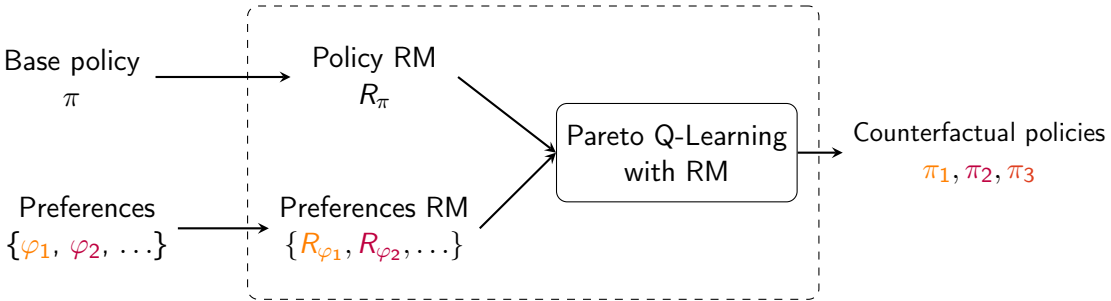
- Base policy in blue
- Counterfactual policy in red

Distance between policies: number of states on which they don't agree

$$d(\pi_1, \pi_2) = \left|\{s \in \mathcal{S} \mid \pi_1(s) \neq \pi_2(s)\}\right|$$

- Given a set of preferences $\Phi = \{\varphi_i\}_i$ and a reference policy π, a policy π' is *counterfactual* w.r.t. Φ and π iff it is the closest policy to π that satisfies all preferences.

Pipeline


Base policy
π
Policy RM
R_π
Preferences
$\{\varphi_1, \varphi_2, \ldots\}$
Preferences RM
$\{R_{\varphi_1}, R_{\varphi_2}, \ldots\}$
Pareto Q-Learning with RM
Counterfactual policies
π_1, π_2, π_3

From LTL_f to reward machines

Co-safety properties: "Something good will eventually happen"

Example: $\Diamond$`treasure`

- Compile φ to a DFA.
- Once an accepting state cannot be left, declare φ satisfied.
- Merge accepting absorbing states into a single u_f; reward $+1$ for transitions into u_f, 0 elsewhere.

Safety properties: "Nothing bad will ever happen"

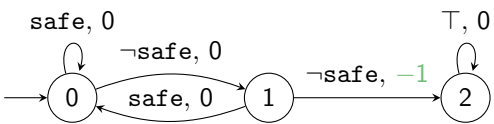
Example: $\Box$`safe`

Dual construction: detect rejecting absorbing states and merge them into a sink. Transitions into the sink yield reward -1.

A *reward machine* (RM) is a finite automaton over $2^{\mathcal{P}}$ that selects which reward function to apply.

LTL_f properties that are (co-)safe can be translated into RMs

Example: $\Box(\neg \texttt{safe} \Rightarrow \bigcirc \texttt{safe})$



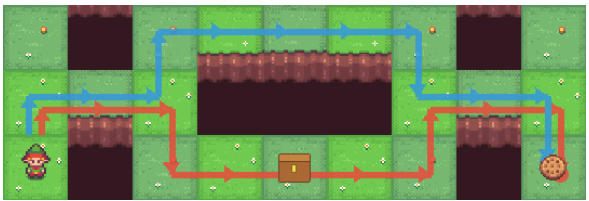
Encoding the reference policy

A single-state RM (= reward function) which tracks agreement with the reference policy π:

$$R_\pi(s,a) = \begin{cases} +1 & \text{if } a = \pi(s) \\ -c & \text{otherwise} \end{cases} \qquad (c \in \mathbb{N})$$

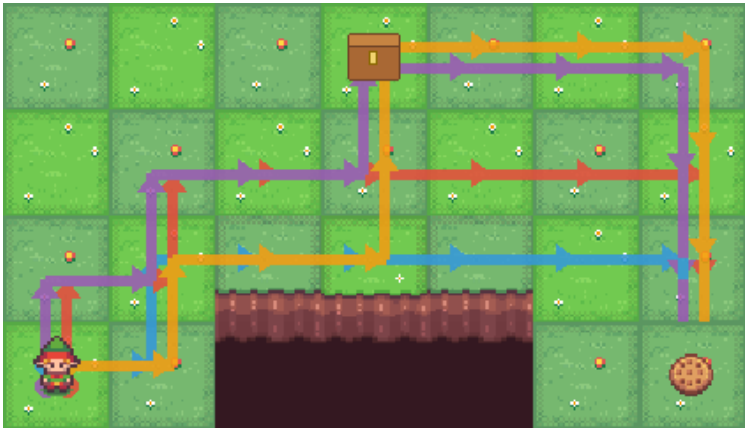
- Dense signal: encourages staying close to π at every step.
- c controls how strongly deviations are penalised.

Cliffwalking



- Preference: $\Diamond$ `treasure`.
- Reference (blue) ignores the treasure.
- Our method returns one extra policy (red) that collects the treasure.
- Both are Pareto optimal.

Non-trivial Pareto Fronts

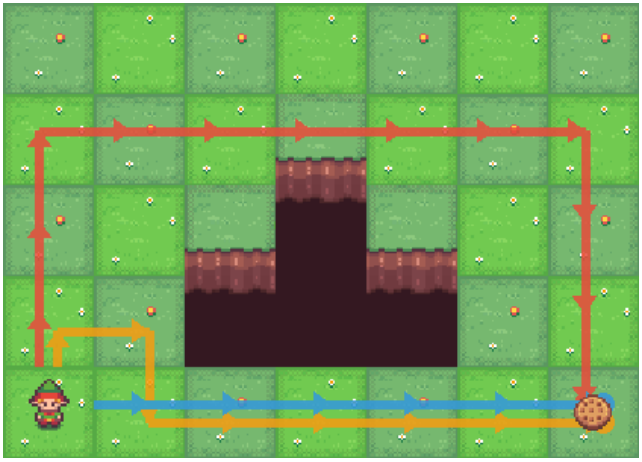


$\varphi_1 = \Diamond\, \texttt{treasure}$

$\varphi_2 = \Box\, \texttt{safe}$

- Base policy in blue (shortest path)
- Multiple policies balances the two objectives

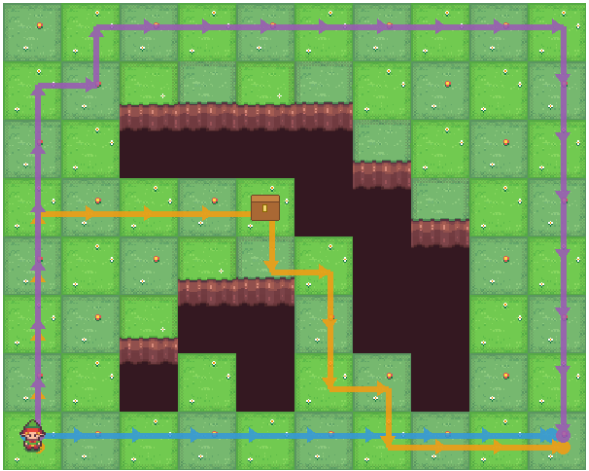
Non-trivial Pareto Fronts



$\varphi_1 = \Diamond(\texttt{safe} \wedge \bigcirc\neg\texttt{safe} \wedge \bigcirc\bigcirc\,\texttt{safe})$

$\varphi_2 = \Box(\neg\texttt{safe} \Rightarrow \bigcirc\texttt{safe})$

Handling conflicting goals



- Treasure sits on an unsafe cell.
- No policy can satisfy both $\Diamond$ `treasure` and $\Box$ `safe`.
- Our method still returns 3 policies: reference + one per preference.
- Q-learning with both rewards struggles to converge.

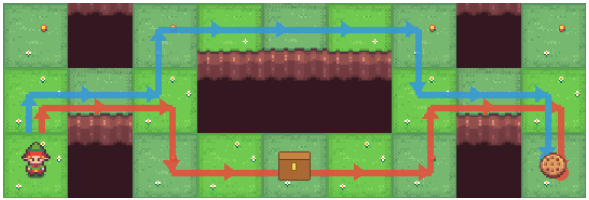
Summarising: reading off the deviations

Replaying π' and comparing each action to what π would do in s_i exposes patterns of agreement and deviation:

state at t	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
π	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
π'	$\rightarrow$	$\rightarrow$	$\uparrow$	$\rightarrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$
R_π	$+1$	$+1$	-1	-1	-1	-1	$+1$	$+1$	$+1$

where $s_{i+1} = \pi'(s_i)$

- Steps 1–2, 7–9: π' **agrees** with π
- Steps 3–6: the **deviation cluster** which is reported to the user

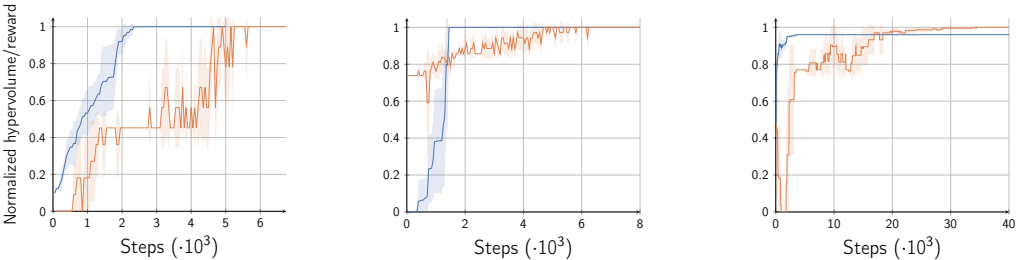


- Preference: $\Diamond$ `treasure`.
- Reference (blue) ignores the treasure.
- Counterfactual policy (red) that collects the treasure.
- Both are Pareto optimal.

Summarising the red policy against the reference π:

step t	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	$\cdots$
π	$\uparrow$	$\rightarrow$	$\uparrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\uparrow$	$\cdots$
π'	$\uparrow$	$\rightarrow$	$\downarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\uparrow$	$\cdots$
R_π	$+1$	$+1$	-1	$+1$	$+1$	$+1$	$+1$	$+1$	$\cdots$

Convergence



- MORL+RM typically converges faster than retraining a dedicated Q-learning agent.
- It returns the whole Pareto set in a single run.

Strengths and limitations

Strengths

- Agnostic to the origin of π (no Q-values required).
- Returns a diverse set of trade-offs in one training run.
- LTL_f preferences are arguably more interpretable than reward functions.

Limitations

- Restricted to safety / co-safety LTL_f formulas.
- Some formulas (e.g. $\Box(x \Rightarrow \Diamond y)$) only approximated via bounded-horizon variants.
- Quality depends on coverage of the input policy.
- Dense step-wise reward for R_π is only a proxy of the original objective.

Take-aways

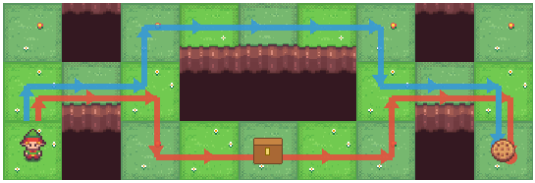
Counterfactual Policies through Temporal Preferences

- Compile π and LTL_f preferences into reward machines.
- Solve the resulting MOMDP with a multi-policy MORL algorithm.
- Obtain a Pareto-optimal set of counterfactual policies.
- Summarise each by highlighting deviations from π.

Future Work

- Tailor a specification language, with a clearer semantics for $\Box/\Diamond$ (soft vs. strict requirements).
- Use Inverse Reinforcement Learning to obtain a richer representation of π.
- Integrate dedicated XRL methods for richer policy comparison.
- User studies on the resulting explanations.

Counterfactual policies



- Base policy in blue
- Counterfactual policy in red

Distance between policies:

$$d(\pi_1, \pi_2) = \left|\{s \in \mathcal{S} \mid \pi_1(s) \neq \pi_2(s)\}\right|$$

Given a set $\Psi = \{\varphi_1, \ldots, \varphi_N\}$ of LTL_f formulas:

$$\mathcal{C}^{\pi}_{\Psi} = \arg\min_{\pi_c} \left\{ d(\pi_c, \pi) \mid \forall \varphi_i \in \Psi,\, \forall s,\, \pi^*_c[s] \models \varphi_i \right\}$$

The CFP-TL Problem

Counterfactual Policies through Temporal Logic Preferences

Input:
- a labelled transition system $\mathcal{T}$,
- a reference policy π on $\mathcal{T}$,
- a set $\Psi = \{\varphi_i\}_{i \leq N}$ of safe / co-safe LTL_f formulas.

Output: a consistent, saturated set Π_{CFP} of policies counterfactual to π w.r.t. subsets of Ψ.

No assumption on the origin of π – only the policy itself is needed.