Reinforcement Learning Roadmap

Phase 6: Safety, Robustness & Explainability

Making RL Agents Trustworthy, Resilient, and Transparent for the Real World

linkedin.com/in/abdullahzahid655 | github.com/abdullahzahid655

February 18, 2026

Roadmap — Phase 6 at a Glance

Why Phase 6?

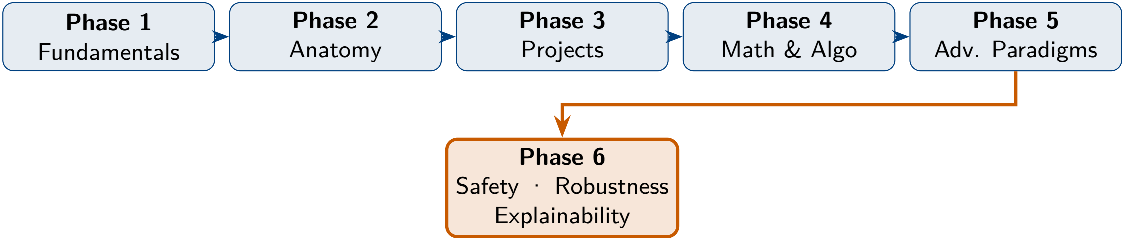
Phases 1–5 taught us *how* to train RL agents.

Phase 6 answers the harder question:

Can we trust them in the real world?

- Safety constraints
- Robustness to attacks
- Explainable decisions

Our Journey So Far



Phase 6 bridges advanced paradigms with **real-world deployment requirements**

Safe Reinforcement Learning

Phase 6: Safety, Robustness & Explainability

What Is Safe RL?

Core Formulation: CMDP

Constrained Markov Decision Process extends the standard MDP by adding cost signals and budget constraints:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \\ \text{s.t.} \quad & \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t c_t\right] \leq b \end{aligned}$$

where c_t is a **cost signal** and b is the **safety budget**.

The Core Tension

Higher reward $\longleftrightarrow$ **Lower safety**

SafeRL learns the Pareto frontier between them.

Real-World Motivation

- **Autonomous Driving**: minimize travel time *while* ensuring low accident probability [Wachi et al., 2024]
- **Power Grids**: optimise electricity production *while* maintaining reliability standards
- **Robotics**: reach target *without* collisions
- **Healthcare**: maximise patient outcome *within* dosage limits

Safe RL Algorithm Families

Lagrangian Methods

Convert constrained problem to unconstrained via dual variable λ:

$$\mathcal{L}(\pi, \lambda) = J^r - \lambda(J^c - b)$$

Examples:

- PPO-Lagrangian
- TRPO-Lagrangian
- PID Lagrangian

Pro: Simple drop-in
Con: Oscillation risk

Trust-Region Methods

Constrain both reward and cost updates within a safe trust region.

$$\pi_{k+1} = \arg\max_{\pi} J^r \text{ s.t. cost} \leq b$$

Examples:

- CPO (Achiam et al., 2017)
- PCPO
- SB-TRPO (2024)

Pro: Monotonic safety
Con: Computationally heavy

Model-Based Methods

Predict cost using a world model, plan safely before execution. **Examples:**

- SafeDreamer (ICLR 2024)
- MOPO + cost model
- CBF-based control

Pro: Near-zero violations
Con: Model errors propagate

Constrained Policy Optimization (CPO)

CPO — Achiam et al., ICML 2017

Updates policy within a trust region *and* respects cost constraints simultaneously:

$$\pi_{k+1} = \arg\max_{\pi} \quad J^r(\pi)$$
$$\text{s.t.} \quad J^c(\pi) \leq b$$
$$D_{KL}(\pi \| \pi_k) \leq \delta$$

Uses first-order Taylor approximation + line-search to solve efficiently.

Key Paper

Achiam, J. et al. (2017). *Constrained Policy Optimization*. ICML 2017. Wachi, A. et al. (2024). *A Survey of Constraint Formulations in Safe RL.*

SafeDreamer: World-Model Safety

Integrates Lagrangian methods into world model planning (Dreamer framework):

1. World model trained from replay buffer
2. Lagrangian planner optimises in latent space
3. Achieves **near-zero cost** on Safety-Gymnasium

Key Paper

Huang, W. et al. (2024). *SafeDreamer: Safe RL with World Models.* ICLR 2024. arXiv:2307.07176

Safe RL: Industry Deployments

Autonomous Driving — Waymo / Tesla

- Constrained RL for lane-keeping and intersection navigation
- Reward: minimize travel time
- Constraints: collision probability $< 10^{-6}$ per mile
- Uses shielding layers as hard safety overrides

Robotic Arm — Industrial Automation

- Adjei et al. (2024): CMDP for arm manipulation avoiding human operators
- Lagrange multiplier λ adapts dynamically to danger proximity
- Published in *Robotics, MDPI 2024*

Power Grid — Energy Management

- Optimise energy dispatch (reward) while satisfying reliability constraints (cost)
- Risk-sensitive CVaR constraints guard against brownouts
- Used in smart-grid pilot programs

Multi-Agent Safe RL — Drone Swarms

- Scal-MAPPO-L (NeurIPS 2024): scalable safe MARL for drone coordination
- Decentralised execution with local constraint satisfaction
- Handles 50+ drones simultaneously

Control Barrier Functions (CBF) for Hard Safety

CBF — Hard Safety Guarantee

A function $h(s)$ is a CBF if the set $\mathcal{C} = \{s : h(s) \geq 0\}$ is **forward-invariant**:

$$\dot{h}(s, a) + \alpha(h(s)) \geq 0 \quad \forall s \in \mathcal{C}$$

Combined with RL: the RL policy proposes actions, CBF *projects* them to the safe set.

- No constraint violations *by construction*
- Works in continuous action spaces
- Used in safety-critical robotics

Safe RL Taxonomy

Approach	Guarantee
Lagrangian	Soft, expectation
CPO / Trust-Region	Soft, monotonic
CBF Shielding	Hard, formal
CMDP offline	Soft, offline data
SafeDreamer	Near-zero, model

Survey

Garcia & Fernández (2015). *Comprehensive Survey on Safe RL*. JMLR 16(1).

Gu et al. (2024). *A Survey of Safe RL*. IEEE TPAMI 2024.

Robust Reinforcement Learning

Phase 6: Safety, Robustness & Explainability

Why Robustness Matters

The Brittleness Problem

DRL agents achieve superhuman performance in controlled environments, but:

- Small observation perturbations *collapse* performance
- A self-driving agent with GPS noise drifts off-road
- Sim-to-real gap invalidates trained policies
- Adversarial attackers can deliberately exploit vulnerabilities

State-Adversarial MDP (SA-MDP)

$$\Omega^{\xi} = (S, A, T, R, \mathcal{X}, O^{\xi})$$

Adversary modifies observations: $O^{\xi}(x_t|s_t)$ Agent must perform well under **worst-case** perturbations.

Types of Adversarial Attacks

1. **Observation Attacks**: perturb agent's state input
   $\rightarrow$ FGSM, PGD variants
2. **Action Attacks**: corrupt agent's actuator output
   $\rightarrow$ NR-MDP framework
3. **Reward Attacks**: manipulate reward signal
   $\rightarrow$ Reward poisoning
4. **Dynamics Attacks**: change environment physics
   $\rightarrow$ Domain-shift attacks
5. **Adversarial Policy**: co-agent manipulates behaviour
   $\rightarrow$ Gleave et al., ICLR 2020

Adversarial Training Framework

Minimax Robust Objective

$$\max_{\pi} \ \min_{\xi \in \Xi} \ \mathbb{E}_{\pi,\xi}[\textstyle\sum_t \gamma^t r_t]$$

Train protagonist π against strongest possible adversary ξ.

ATLA (Alternating Training of Learned Adversary):

1. Train optimal adversary ξ^* against current π
2. Train π against current ξ^*
3. Alternate until convergence

Key Papers

Zhang et al. (2020). *Robust DRL against Adversarial Perturbations.* NeurIPS 2020.
Schott et al. (2024). *Robust DRL Through Adversarial Attacks and Training.* arXiv:2403.00420

Robustness Techniques

Training-Time:

- Domain Randomisation
- Adversarial Observation Training
- Noise Augmentation (NA-PPO)
- RADIAL-RL: adversarial loss regularisation

Test-Time:

- Certified robustness (CROP)
- Ensemble voting
- Input preprocessing / detection

Evaluation:

- GWC (Greedy Worst-Case Reward)
- Attack-agnostic benchmarks

Domain Randomisation: Sim-to-Real

Core Idea

Randomise environment parameters during training so the policy learns to generalise:

- Friction, mass, gravity coefficients
- Sensor noise levels
- Lighting & textures (for vision)
- Actuator delays and latency

Policy sees distribution $p(\xi)$ of environments $\Rightarrow$ robust to real-world variations.

Notable Work

OpenAI (2019): *Dexterous In-Hand Manipulation* — robotic hand solving Rubik's cube via massive domain randomisation.
Chen et al. (2024). *Adversarial Domain Randomization for Dual-UAV Cooperation.*

Sim-to-Real Examples

Boston Dynamics Atlas:

- Trained in simulation with randomised terrain
- Zero-shot transfer to physical robot

Industrial Assembly Robots:

- Part orientation variance
- Tool wear randomisation
- Successfully deployed in BMW factories

UAV Drone Swarms:

- Wind disturbance randomisation
- Communication latency variance
- NeurIPS 2024: Scal-MAPPO-L

RADIAL-RL: Certified Adversarial Robustness

RADIAL-RL Framework

Trains agents with **adversarial loss** as a regulariser:

$$\mathcal{L}_{total} = \mathcal{L}_{RL} + \lambda_{adv} \cdot \mathcal{L}_{adv}$$

where $\mathcal{L}_{adv}$ is the worst-case loss over the l_p-ball perturbation set.

Compatible with: DQN, A3C, PPO
Tested on: Atari, MuJoCo, ProcGen

Paper

Oikarinen et al. (2021). *Robust DRL Through Adversarial Loss.* NeurIPS 2021. `https://github.com/tuomaso/radial_rl_v2`

Benchmark Results (Pong)

Method	**Clean**	**Under Attack**
Vanilla DQN	21	−21
SA-DQN	21	21
RADIAL-DQN	21	20

SA-DQN / RADIAL-DQN maintain full performance under PGD attacks that *completely destroy* vanilla DQN.

Explainable Reinforcement Learning (XRL)

Phase 6: Safety, Robustness & Explainability

The Black-Box Problem in RL

Why Is RL Hard to Explain?

- Policies are **deep neural networks** — millions of parameters
- Decisions depend on **sequences of states** (temporal credit)
- Emergent strategies arise from complex reward shaping
- Standard XAI (LIME, SHAP) was designed for supervised learning

Definition: XRL

*"Explainable RL (XRL) is an emerging subfield that aims to elucidate the decision-making process of RL agents, enabling practitioners to understand **what** agents will do and **why**."* [Milani et al., ACM 2023]

Stakeholder Questions XRL Answers

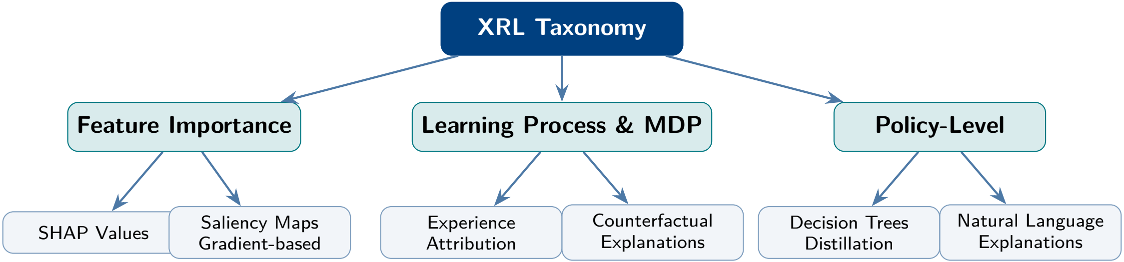
1. **Why** did the agent take action a in state s?
2. **What** features matter most to the policy?
3. **When** does the agent fail or behave unexpectedly?
4. **How** will the policy behave on unseen states?
5. **What** subgoals is the agent pursuing?

Surveys

Bekkemoen, Y. (2024). *XRL: Systematic Literature Review and Taxonomy.* Machine Learning 113.

Milani et al. (2023). *XRL: A Survey and Comparative Review.* ACM Comput. Surv.

XRL Taxonomy



Based on Milani et al. (2023) ACM Computing Surveys taxonomy.

SHAP for RL: Shapley Values

Shapley Value Attribution

Assign credit to each feature i for the Q-value:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[v(S \cup \{i\}) - v(S) \right]$$

$\phi_i > 0$: feature *increased* action value
$\phi_i < 0$: feature *decreased* action value

Application: XRL Governance

Pakina et al. (2024). *AI Governance via XRL for Adaptive Cyber Deception in Zero-Trust Networks.* JISEM 2024.

SHAP raised decision transparency from **0%** to **94%**.

SHAP in RL Pipeline

1. Train DQN / PPO agent normally
2. Wrap Q-network with SHAP explainer
3. For each state s, compute ϕ_i for all features
4. Visualise as bar plot or heatmap
5. Audit: do top features make sense?

Libraries:
`shap`, `captum` (PyTorch)

Policy Distillation: Interpretable Surrogates

Core Idea

Distil a trained DNN policy into a simpler, interpretable model:

1. Train a high-performing DNN policy π_{DNN}
2. Generate a large dataset of $(s, \pi_{DNN}(s))$ pairs
3. Fit an interpretable model: decision tree, linear model, rule list
4. Use surrogate for deployment & auditing

Interpretable Surrogates

Surrogate	Fidelity	Interpretability
Linear Model	Medium	Very high
Decision Tree	Medium	High
Rule List	Medium	Very high
Shallow NN	High	Low
Prototype	High	Medium

Research

Dhebar et al. (2024). *Toward Interpretable-AI Policies Using Evolutionary Nonlinear Decision Trees.* IEEE Trans. Cybern.

Beechey et al. (2023). *Explaining RL with Shapley Values.* ICML 2023.

XRL-SHAP-Cache

Hu et al. (2024, Springer). Combined DRL + SHAP for **intelligent edge service caching** in 5G CDNs — decisions fully auditable by network engineers.

Counterfactual Explanations in XRL

What Are Counterfactuals?

*"What **minimal change** to state s would cause the agent to take a **different action**?"*

$$\mathsf{s}^{CF} = \arg\min_{s'} \|s' - s\| \text{ s.t. } \pi(s^{CF}) \neq \pi(s)$$

Counterfactuals provide **actionable** explanations — they tell users what *would have been different.*

Research

Amitai et al. (2024). *Explaining RL Agents through Counterfactual Action Outcomes.* AAAI 2024.

GANterfactual-RL: visual counterfactuals for Atari agents (2023).

Healthcare XRL Example

Clinical Decision Support:

- RL optimises treatment dosing
- Doctor asks: *Why did you recommend dose X?*
- SHAP shows: *creatinine level was the deciding feature*
- Counterfactual: *if creatinine* < 1.2, *dose would be Y*

Medical XRL

Ali et al. (2024). *XRL for Alzheimer's Disease Progression Prediction: SHAP-based Approach.* AAAI XAI4DRL Workshop 2024.

Connecting Phase 5 + Phase 6

Phase 6: Safety, Robustness & Explainability

Phase 5 × Phase 6: Synergies

Safe MARL

Multi-agent systems with safety constraints:

- NeurIPS 2024: **Scal-MAPPO-L** — scalable safe MARL for drone swarms
- MACPO: Multi-Agent Constrained Policy Optimisation
- Challenge: individual vs shared safety constraints

Explainable HRL

Hierarchical policies are naturally more interpretable:

- High-level goal is human-readable (*"go to kitchen"*)
- Low-level actions can be audited per subgoal
- Counterfactuals at task-decomposition level

Robust Meta-Learning

Meta-RL + robustness to task distribution shifts:

- Adapt quickly to new tasks without losing safety
- Distributionally robust MAML
- Offline safe meta-RL

Safe Offline RL

FISOR (ICLR 2024): combines offline RL + hard safety constraints:

- Feasibility-guided decoupled learning
- Hamilton-Jacobi reachability for safe region detection
- Best safety on DSRL benchmark

Practical Resources

Phase 6: Safety, Robustness & Explainability

Key Research Papers — Phase 6

Safe RL

- Achiam et al. (2017). *CPO.* ICML.
- Garcia & Fernández (2015). *Survey on Safe RL.* JMLR.
- Huang et al. (2024). *SafeDreamer.* ICLR. arXiv:2307.07176
- Wachi et al. (2024). *Survey on Constraint Formulations.* arXiv:2402.02025
- Liu et al. (2024). *FISOR: Feasibility-guided Safe Offline RL.* ICLR.
- NeurIPS 2024. *Scal-MAPPO-L.* Safe Multi-Agent RL.

Robust RL

- Zhang et al. (2020). *SA-DQN.* NeurIPS Spotlight.
- Oikarinen et al. (2021). *RADIAL-RL.* NeurIPS.
- Schott et al. (2024). *Survey: Adversarial Attacks & Training.* arXiv:2403.00420
- Liu et al. (2024). *Safe offline RL + distributional robustness.* NeurIPS.

Explainable RL

Bekkemoen (2024). *XRL Systematic Literature Review.* Machine Learning 113. Milani et al. (2023). *XRL Survey.* ACM Comput. Surv. Beechey et al. (2023). *Explaining RL with Shapley Values.* ICML. Pakina et al. (2024). *AI Governance via XRL.* JISEM. Amitai et al. (2024). *Counterfactual Action Outcomes.* AAAI.

Libraries, Benchmarks & Tools

Safe RL

- **Safety-Gymnasium**: unified safe RL benchmark
- **DSRL**: offline safe RL datasets
- **OmniSafe**: safe RL algorithm library
- **safe-control-gym**: CBF + RL
- **SafeRL-kit**: reference implementations

Robust RL

- **SA-DQN codebase**: GitHub (chenhongge)
- **RADIAL-RL**: GitHub (tuomaso/radial_rl_v2)
- **MuJoCo**: physics engine for testing
- **ProcGen**: procedurally generated benchmark
- **RobustBench**: adversarial robustness leaderboards

Explainability

- **SHAP**: `pip install shap`
- **Captum** (PyTorch): saliency, IG, SHAP
- **Gymnasium**: policy replay
- **Weights & Biases**: training transparency
- **ProtoX**: prototype-based XRL

Hands-On Projects for Phase 6

Project 1: Safe CartPole / LunarLander

1. Define a cost: pole angle $>$ threshold $=$ unsafe
2. Implement PPO-Lagrangian from scratch
3. Compare reward vs. constraint violation trade-off
4. Visualise Lagrange multiplier λ over training
5. **Extension**: add CBF safety layer

Project 2: Robust DQN on Atari

1. Train standard DQN on Pong
2. Apply FGSM observation attack — watch it fail
3. Implement SA-DQN (adversarial training)
4. Measure GWC reward before vs. after
5. Plot robustness vs. ϵ budget curve

Project 3: XRL Dashboard (This Series!)

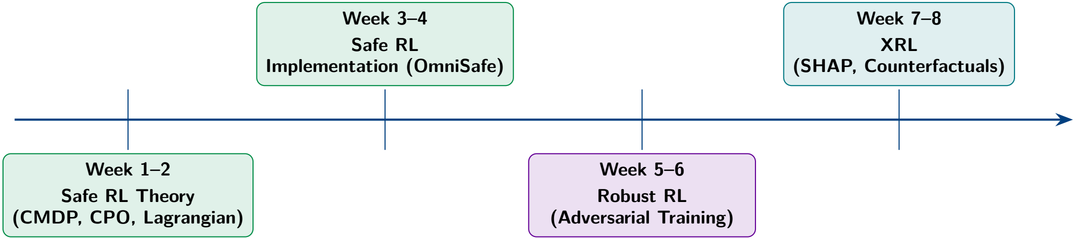
1. Train DQN on CartPole / Taxi-v3
2. Apply SHAP to Q-network at each step
3. Visualise top-3 features per action
4. Generate counterfactual states
5. Distil policy into a decision tree
6. **Build an explainability dashboard**

$\rightarrow$ *This is the integrated project in our Jupyter notebook!*

Project 4: Safe MARL Drone

Implement safe cooperative navigation using MACPO in PettingZoo — agents reach goals without collisions.

8-Week Study Plan — Phase 6



Weeks 1–2: Safe RL Theory

- Read: Garcia & Fernández survey; CPO paper
- Understand CMDPs and Lagrangian duality
- Run Safety-Gymnasium starter examples

Weeks 7–8: XRL

- Read: Milani et al. ACM survey on XRL
- Implement SHAP on a trained DQN
- Build Project 3: XRL Dashboard

Integrated RL Project (Phase 5 + 6)

Phase 6: Safety, Robustness & Explainability

Integrated Project: Safe & Explainable RL Agent

Project Overview

Environment: OpenAI Gymnasium `CartPole-v1` (extended with safety cost)

Phase 5 contributions:

- Offline RL: pre-train from logged CartPole data
- Meta-Learning: fast-adapt to perturbed pole lengths

Phase 6 contributions:

- Safety: PPO-Lagrangian with angle cost
- Robustness: adversarial noise on observations
- Explainability: SHAP + decision tree distillation

Jupyter Notebook Structure

1. **Setup**: Install deps, env creation
2. **Baseline DQN**: train standard agent
3. **Offline RL**: pre-training from replay
4. **Safe RL**: add cost + PPO-Lagrangian
5. **Robust RL**: adversarial attack + SA-DQN
6. **XRL**: SHAP attribution plots
7. **Policy Distillation**: decision tree
8. **Dashboard**: compare all agents

Full code available at:
`github.com/abdullahzahid655`

Summary & Next Steps

Phase 6: Safety, Robustness & Explainability

Phase 6 Summary

What We Covered

1. **Safe RL**: CMDPs, Lagrangian methods, CPO, SafeDreamer, CBFs
2. **Robust RL**: adversarial attacks, SA-MDP, RADIAL-RL, domain randomisation
3. **XRL**: SHAP, saliency, counterfactuals, policy distillation, taxonomy
4. **Synergies**: safe MARL, robust meta-RL, safe offline RL
5. **Industry**: Waymo, Boston Dynamics, power grids, healthcare

Key Insights

- Safety $\neq$ Robustness $\neq$ Explainability — each addresses a different deployment risk
- All three are needed for real-world deployment
- Combining with Phase 5 paradigms unlocks the most powerful systems
- Active research area — new papers weekly

Coming Next — Phase 7: Model-Based RL & World Models

Learning dynamics models, Dyna, Dreamer, MuZero, planning with uncertainty — the key to sample efficiency.

Thank You!

Questions & Discussion

Follow the RL Roadmap Series:

linkedin.com/in/abdullahzahid655   github.com/abdullahzahid655

"An unsafe, brittle, or opaque AI is not truly intelligent — it is merely lucky."

References I

- Achiam, J. et al. (2017). *Constrained Policy Optimization.* ICML.
- Wachi, A. et al. (2024). *A Survey of Constraint Formulations in Safe RL.* arXiv:2402.02025.
- Huang, W. et al. (2024). *SafeDreamer: Safe RL with World Models.* ICLR. arXiv:2307.07176.
- Milani, S. et al. (2023). *XRL: A Survey and Comparative Review.* ACM Comput. Surv.
- Bekkemoen, Y. (2024). *XRL: Systematic Literature Review.* Machine Learning 113.
- Zhang, H. et al. (2020). *Robust DRL against Adversarial Perturbations.* NeurIPS 2020 (Spotlight).
- Oikarinen, T. et al. (2021). *RADIAL-RL.* NeurIPS 2021.
- Schott, L. et al. (2024). *Robust DRL: Adversarial Attacks and Training Survey.* arXiv:2403.00420.
- Liu, Z. et al. (2024). *FISOR: Feasibility-guided Safe Offline RL.* ICLR 2024.
- Pakina, A. et al. (2024). *AI Governance via XRL for Cyber Deception.* JISEM 2024.
- Amitai, Y. et al. (2024). *Explaining RL Agents via Counterfactual Action Outcomes.* AAAI 2024.

References II

- Adjei, P. et al. (2024). *Safe RL for Arm Manipulation with CMDP.* Robotics, MDPI 13(4).
- Beechey, D. et al. (2023). *Explaining RL with Shapley Values.* ICML 2023.