

A Survey of Safe Reinforcement Learning and Constrained MDPs: A Technical Survey on Single-Agent and Multi-Agent Safety

Ankita Kushwaha¹, Kiran Ravish¹, Preeti Lamba¹, and Pawan Kumar¹

¹International Institute of Information Technology, Hyderabad

May 26, 2025

Abstract

Safe Reinforcement Learning (SafeRL) is the subfield of reinforcement learning that explicitly deals with safety constraints during the learning and deployment of agents. This survey provides a mathematically rigorous overview of SafeRL formulations based on Constrained Markov Decision Processes (CMDPs) and extensions to Multi-Agent Safe RL (SafeMARL). We review theoretical foundations of CMDPs, covering definitions, constrained optimization techniques, and fundamental theorems. We then summarize state-of-the-art algorithms in SafeRL for single agents, including policy gradient methods with safety guarantees and safe exploration strategies, as well as recent advances in SafeMARL for cooperative and competitive settings. Additionally, we propose five open research problems to advance the field, with three focusing on SafeMARL. Each problem is described with motivation, key challenges, and related prior work. This survey is intended as a technical guide for researchers interested in SafeRL and SafeMARL, highlighting key concepts, methods, and open future research directions.

Contents

1	Introduction	2
2	Related Work	3
3	Safe Reinforcement Learning and Constrained MDPs: Foundations	3
3.1	Markov Decision Processes (MDPs)	4
3.2	Constrained Markov Decision Processes (CMDPs)	4
3.3	Constraint Types and Safety Specifications	6
3.4	Theoretical Results	12
4	State-of-the-Art Methods in SafeRL and SafeMARL	13
4.1	Lagrangian-based Policy Optimization	14
4.2	Safety Shields and Action Correction	15
4.3	Risk-Sensitive and Distributional Methods	15
4.4	Safe Multi-Agent Reinforcement Learning (SafeMARL)	16
5	Open Research Challenges and Future Directions	18
6	Conclusion	20

1 Introduction

Reinforcement learning (RL) has achieved remarkable success in domains such as games, robotics, and autonomous systems. However, when deploying RL in real-world *safety-critical* applications (e.g., autonomous driving, healthcare, robotics), it is essential to ensure that the learning agent avoids catastrophic failures or unsafe behaviors Amodei et al. [2016], Garcia and Fernandez [2015]. **Safe Reinforcement Learning (SafeRL)** addresses this need by augmenting standard RL objectives with safety considerations, typically in the form of constraints on the agent’s behavior or environment outcomes.

Definition 1.1. *The goal in SafeRL is to maximize performance (cumulative reward) while satisfying safety constraints during training and deployment.*

A common framework for SafeRL is the **Constrained Markov Decision Process (CMDP)** introduced by Altman [1999]. In a CMDP, an agent seeks to maximize expected return subject to one or more constraints (e.g., bounds on certain costs or probabilities of unsafe events). This framework allows formalizing safety requirements as mathematical constraints and provides tools from constrained optimization and control theory to enforce them. SafeRL algorithms often leverage CMDP theory to find policies that respect constraints (at least approximately) while learning efficiently. SafeRL has gained significant attention in recent years. Early work in SafeRL explored modifications of the RL objective to encode risk or safety (e.g., worst-case guarantees Heger [1994], risk-sensitive criteria Borkar [2002], or probability of failure constraints Geibel and Wysotski [2005]). More recent approaches explicitly enforce constraints during learning using techniques like Lagrange multipliers, trust-region methods, or safety monitors. There have been comprehensive surveys of SafeRL (e.g., Garcia and Fernandez [2015]) and increasing theoretical study of constrained RL algorithms Achiam et al. [2017], Chow et al. [2018a]. An emerging frontier is **Multi-Agent Safe Reinforcement Learning (SafeMARL)**, which considers multiple agents learning and interacting under safety constraints. SafeMARL is crucial for applications like co-ordinated robotics, drone swarms, and autonomous driving with multiple vehicles, where safety conditions involve interactions among agents. SafeMARL introduces additional challenges such as coordinating safety in a team, handling the coupling of constraints across agents, and new solution concepts (like safe equilibria Ganzfried [2022] in competitive settings). While single-agent SafeRL is relatively well-studied, SafeMARL remains a young research area with many open problems ElSayed-Aly et al. [2021], Gu et al. [2023]. This survey provides

- A rigorous introduction to SafeRL formulations based on CMDPs, including mathematical definitions and theorems.
- A review of state-of-the-art SafeRL methods for single agents, and their extensions to multi-agent scenarios (SafeMARL), highlighting major algorithms and theoretical guarantees.
- A discussion of related work and different perspectives on safety in RL (e.g., risk-sensitive RL, robust RL, safe exploration techniques).
- Five open research problems that we believe are important for advancing SafeRL and SafeMARL. Three of these focus specifically on challenges in SafeMARL.

Our target audience is researchers familiar with fundamental RL concepts who seek a deeper understanding of how to incorporate safety in RL. We assume knowledge of basic RL (Markov decision processes, policy optimization, etc.) and provide definitions and notation for SafeRL topics. We believe that by the end of this paper, the reader should be equipped with the theoretical background of CMDPs, knowledge of leading algorithms in SafeRL/SafeMARL, and insight into promising research directions in this field.

2 Related Work

SafeRL has been surveyed and reviewed from multiple angles. Garc’ia and Fern’andez Garcia and Fernandez [2015] provide an earlier comprehensive survey of SafeRL methods up to 2015, categorizing approaches into modifications of the optimality criterion (e.g., constrained or risk-sensitive objectives) and modifications of the exploration process (e.g., using external knowledge or risk metrics to guide learning). They classify safety criteria into four groups:

- *Constrained criteria* – optimization with explicit constraints on policies Geibel and Wysotski [2005],
- *Worst-case (robust) criteria* – optimize the minimal possible return under adversarial conditions Heger [1994],
- *Risk-sensitive criteria* – incorporate risk measures like variance or CVaR (Conditional Value-at-Risk) into the objective Borkar [2002], Tamar et al. [2015],
- *Others* – e.g., criteria based on higher moments or probability of ruin.

Our survey focuses primarily on the constrained criterion approach (CMDPs), which has become the prevalent formalism for SafeRL in recent years. Since 2015, the field has advanced with new algorithms and theoretical results. Recent reviews such as Wachi *et al.* Wachi et al. [2024] examine various formulations of safety constraints (e.g., how constraints are represented and enforced) and draw connections between them. Another forthcoming survey by Gu *et al.* Gu et al. [2024] provides an extensive review of SafeRL methods, theory, and applications, reflecting the growing maturity of the field. These works indicate an increasing interest in unifying SafeRL concepts and developing a systematic understanding of safety constraint representations and their implications. On the multi-agent side, SafeMARL has been less surveyed due to its emergent status. Gu *et al.* Gu et al. [2023] investigate safe multi-robot control tasks and propose algorithms like Multi-Agent Constrained Policy Optimization (MACPO). Some recent papers introduce safe multi-agent learning algorithms or frameworks ElSayed-Aly et al. [2021], Gu et al. [2023], Zhang et al. [2024], but a comprehensive survey of SafeMARL is still lacking. Our work contributes by reviewing both single-agent and multi-agent safe RL in one document and highlighting SafeMARL-specific challenges. Other related areas include **robust RL** (handling model uncertainty or adversarial disturbances) and **reward hacking / alignment** (ensuring the specified reward leads to intended safe behavior). While robust RL (e.g., solving worst-case MDPs) and SafeRL share some techniques (like min-max optimization), they address different problem formulations (uncertainty vs. explicit constraints). Similarly, reward specification and alignment problems are complementary to SafeRL: one can combine learned reward shaping with SafeRL constraints to yield agents that both seek correct objectives and stay safe Amodei et al. [2016], Achiam et al. [2017]. We touch upon these connections where relevant. In summary, our survey builds upon and extends prior work by providing a focused treatment of CMDP-based SafeRL and the novel area of SafeMARL, presented in a rigorous yet accessible manner for researchers.

3 Safe Reinforcement Learning and Constrained MDPs: Foundations

In this section, we introduce the theoretical foundations of SafeRL with an emphasis on Constrained Markov Decision Processes (CMDPs). We present formal definitions, notation, and key mathematical results that underpin SafeRL algorithms. We also discuss how safety constraints are formulated and how they can be tackled using constrained optimization techniques in an RL context.

3.1 Markov Decision Processes (MDPs)

We begin with the standard Markov Decision Process (MDP) formulation of an RL problem. An MDP is defined by the tuple $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where

- \mathcal{S} is a (finite or continuous) set of states.
- \mathcal{A} is a set of actions available to the agent.
- $P(s'|s, a)$ is the transition probability function (Markovian dynamics), giving the distribution over next states s' when action a is taken in state s .
- $r(s, a)$ is a reward function (or $r(s, a, s')$ including next state, depending on context) giving a scalar reward for executing action a in state s .
- $\gamma \in [0, 1]$ is a discount factor that weights immediate vs. future rewards (with $\gamma < 1$ typically for infinite-horizon problems).

A (stationary) **policy** π is a mapping from states to a distribution over actions. We denote $\pi(a|s)$ as the probability of taking action a in state s under π .

The value function for a policy π is

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s \right],$$

the expected cumulative discounted reward starting from state s and following π . The goal in standard RL is to find an optimal policy π^* maximizing $V^\pi(s)$ for all s (or maximizing a specific initial state or distribution performance). Equivalently, one maximizes the **return** $J(\pi) = \mathbb{E}_{s_0 \sim \rho} [V^\pi(s_0)]$ for some initial state distribution ρ . In unconstrained RL, π^* solves $\max_\pi J(\pi)$.

3.2 Constrained Markov Decision Processes (CMDPs)

A Constrained Markov Decision Process extends an MDP with the concept of *costs* (or negative rewards) and associated constraints. Formally, a CMDP can be defined as:

$$M_C = (S, A, P, r, \{c^{(i)}\}_{i=1}^m, \gamma),$$

where $r(s, a)$ is the primary reward as before, and $c^{(i)}(s, a)$ for $i = 1, \dots, m$ are m **cost functions** (or penalty functions) encoding the aspects of the task we want to constrain. Each cost function usually corresponds to a particular notion of “unsafe” behavior or resource usage that should be limited. For example, $c^{(1)}(s, a)$ might be an indicator of entering an unsafe state or a measure of damage/risk at state s . A policy π in a CMDP has not only a reward return $J(\pi) = \mathbb{E}_\pi [\sum_t \gamma^t r(s_t, a_t)]$, but also a cost return for each cost function:

$$J_c^{(i)}(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t c^{(i)}(s_t, a_t) \right].$$

The safe RL objective can be posed as a **constrained optimization problem**

$$\text{maximize}_\pi \quad J(\pi) = \mathbb{E}_\pi \left[\sum t \gamma^t r(s_t, a_t) \right], \quad \text{subject to} \quad J_{c^{(i)}}(\pi) \leq d_i, \quad i = 1, 2, \dots, m, \quad (1)$$

where d_i is a specified threshold for the i -th cost (safety limit). The set

$$\Pi_{\text{safe}} = \{\pi \mid J_{c^{(i)}}(\pi) \leq d_i, \forall i\}$$

is called the **feasible policy set**. We assume this set is non-empty (the constraints are attainable). Problem (1) is the standard formulation of SafeRL as a CMDP optimization problem Altman

[1999]. It is a constrained Markov decision problem which, in principle, can be solved via dynamic programming or linear programming if the model is known and state-action spaces are small. Eitan Altman's foundational work Altman [1999] established that for finite CMDPs, there exists an optimal policy that is stationary (time-independent) and, if multiple constraints are present, possibly stochastic (randomized). Intuitively, sometimes a mixture of actions is required to exactly satisfy multiple constraints: a deterministic policy might violate a constraint, whereas a stochastic policy can blend strategies to meet the constraint bounds exactly.

Lagrangian formulation: A common theoretical approach to solve CMDPs is to form the Lagrangian of (1). Introduce Lagrange multipliers $\lambda = (\lambda_1, \dots, \lambda_m) \geq 0$ for the m constraints. The Lagrangian for policy π is

$$\mathcal{L}(\pi, \lambda) = J(\pi) + \sum_{i=1}^m \lambda_i (d_i - J_c^{(i)}(\pi)).$$

We can rearrange $\mathcal{L}(\pi, \lambda) = J(\pi) - \sum_i \lambda_i J_c^{(i)}(\pi) + \sum_i \lambda_i d_i$. Often it is written as $J(\pi) - \sum_i \lambda_i (J_c^{(i)}(\pi) - d_i)$ or $J(\pi) - \sum_i \lambda_i J_c^{(i)}(\pi)$ up to constants, since $\sum_i \lambda_i d_i$ does not depend on π . For a fixed λ , the term

$$J(\pi) - \sum_i \lambda_i J_c^{(i)}(\pi) = \mathbb{E}_\pi \left[\sum_t \gamma^t (r(s_t, a_t) - \sum_i \lambda_i c^{(i)}(s_t, a_t)) \right].$$

This suggests defining a *penalized reward*

$$r_\lambda(s, a) = r(s, a) - \sum_{i=1}^m \lambda_i c^{(i)}(s, a).$$

For any $\lambda \geq 0$, we can compute

$$\pi^*(\lambda) = \arg \max_{\pi} \mathcal{L}(\pi, \lambda) = \arg \max_{\pi} \mathbb{E}_\pi \left[\sum_t \gamma^t r_\lambda(s_t, a_t) \right],$$

which is the optimal policy for the MDP with reward r_λ . In other words, $\pi^{(\lambda)}$ is the unconstrained optimal policy if we treat $-\lambda_i$ as a weight (penalty) for cost $c^{(i)}$. The dual function is

$$g(\lambda) = \max_{\pi} \mathcal{L}(\pi, \lambda) = J(\pi^{(\lambda)}) + \sum_i \lambda_i (d_i - J_c^{(i)}(\pi^{(\lambda)})). \quad (2)$$

We then minimize $g(\lambda)$ over $\lambda \geq 0$ to find the best multipliers

$$\lambda^* = \arg \min_{\lambda \geq 0} g(\lambda).$$

Under certain conditions (convexity or linearity of the CMDP problem), strong duality holds and solving the dual yields the primal optimum Altman [1999], Achiam et al. [2017]. The optimal policy π^* for the CMDP is then $\pi^*(\lambda^*)$ (or a mixture of policies if needed when the optimum is not unique). The Lagrangian perspective is very useful in SafeRL for the following reasons

- It leads to **Lagrange multiplier methods** for safe RL, where one maintains estimates of λ_i during learning and adjusts them based on constraint violations. Many algorithms in practice (Section 4) use this primal-dual approach.
- It gives insight into how costs trade off with reward: λ_i can be interpreted as the “price” of violating constraint i . A high λ_i at optimum means the agent sacrifices a lot of reward to reduce cost i .
- The gradient of $g(\lambda)$ can be derived as $\nabla_{\lambda_i} g(\lambda) = d_i - J_c^{(i)}(\pi^*(\lambda))$. This leads to a gradient descent update: $\lambda_i \leftarrow \lambda_i + \alpha (J_c^{(i)}(\pi) - d_i)$ which intuitively increases the penalty λ_i if constraint i is violated ($J_c^{(i)} > d_i$) and decreases it if the constraint is satisfied with slack.

Linear programming solution: For finite-state CMDPs, an alternative formulation is via occupancy measures and linear programming. One can define variables $x(s, a)$ representing the discounted visitation frequency of state-action pair (s, a) under a stationary policy. The constraints of an optimal occupancy measure include flow conservation (infinite-horizon occupancy distribution) and positivity.

The total expected discounted reward under a stationary policy π is defined as:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

The occupancy measure $x(s, a)$ represents the discounted visitation frequency of state-action pairs under policy π

$$x(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a \mid \pi)$$

which is the expected discounted number of times that the agent visits state s and takes action a .

By unrolling the expectation, the expected return can be rewritten in terms of the occupancy measure:

$$J(\pi) = \sum_{s, a} x(s, a) r(s, a)$$

This expression is the key to formulating the CMDP as a linear program since the objective becomes linear in $x(s, a)$. Furthermore, the expected cumulative cost constraints can be similarly written as

$$J_{c^{(i)}}(\pi) = \sum_{s, a} x(s, a) c^{(i)}(s, a) \leq d_i, \quad \forall i = 1, \dots, m$$

which are also linear in $x(s, a)$. This linear structure is crucial as it allows the CMDP optimization problem to be expressed as a linear program (LP).

The CMDP can then be written as a linear program:

$$\begin{aligned} & \max_{x(s, a) \geq 0} \sum_{s, a} x(s, a) r(s, a) \\ \text{s.t. } & \sum_{s, a} x(s, a) c^{(i)}(s, a) \leq d_i, \quad i = 1, \dots, m, \\ & \sum_a x(s, a) = (1 - \gamma) \rho(s) + \gamma \sum_{s', a'} P(s|s', a') x(s', a'), \forall s, \end{aligned}$$

where $\rho(s)$ is the starting state distribution. This linear program can be solved efficiently for moderate state-action sizes and yields an optimal (potentially stochastic) policy for the CMDP Altman [1999]. While model-based and not directly applicable to large-scale problems, this approach provides theoretical validation that CMDPs are solvable optimally and also serves as a basis for certain planning algorithms in safe RL.

3.3 Constraint Types and Safety Specifications

The formulation above uses expected cumulative costs as constraints. This is a flexible and popular choice in SafeRL research, but it is worth noting other types of constraints that have been considered

1. **Instantaneous constraints:** instead of long-term expected cost, one could require $c(s_t, a_t) \leq d$ at every time step t (almost surely). This is a stricter requirement (no violations at all). Such hard constraints are challenging for learning, and often enforced via external mechanisms (like safety filters). In CMDP theory, instantaneous constraints can be encoded by making any violation transition to an absorbing failure state with heavy penalty.

Examples of Instantaneous Constraints in Safe RL

Instantaneous constraints refer to safety requirements that must hold *at every time step* during the agent’s execution, rather than only in expectation over a trajectory. Below are typical examples of such constraints arising in real-world applications.

Robotics — Torque or Force Limits: Robotic manipulators and mobile robots have strict actuator limits. A common constraint is $\|\tau_t\| \leq \tau_{\max}$, where τ_t is the torque vector applied at time t . Exceeding these limits even once can cause irreversible hardware damage. Therefore, the torque constraint must hold at every step.

Autonomous Driving — Collision Avoidance: Autonomous vehicles must avoid collisions at all times. This is often modeled as a minimum distance constraint, $\text{distance}(s_t) \geq d_{\text{safe}}$, where d_{safe} is a safety margin. Unlike reward penalties for collisions, instantaneous constraints aim to ensure that no collision ever occurs, even during learning.

Aerial Vehicles (Drones) — Altitude Constraints: Drones often operate within restricted altitude corridors, leading to constraints of the form $z_{\min} \leq z_t \leq z_{\max}$. Exceeding altitude boundaries may result in collisions with terrain (if $z_t < z_{\min}$) or violation of airspace regulations (if $z_t > z_{\max}$). Such constraints must be enforced at all times.

Medical Applications — Dose Limits in Treatment Planning: In adaptive radiation therapy or drug administration, instantaneous dosage constraints are essential. The instantaneous constraint may take the form $\text{dose}_t \leq d_{\max}$, limiting the maximum dose administered at each step to prevent severe side effects.

Power Systems — Voltage and Current Limits: Power grids are subject to operational safety limits such as $V_t \in [V_{\min}, V_{\max}]$ for voltage levels or similar constraints on current. Violations could cause system instability, equipment damage, or even large-scale blackouts. Safe control must respect these constraints instantaneously.

Industrial Process Control — Pressure Limits: In chemical plants, nuclear reactors, and manufacturing systems, pressure constraints of the form $p_t \leq p_{\max}$ are typical. Exceeding pressure thresholds even once may lead to catastrophic failures such as explosions or hazardous material leaks.

These types of instantaneous constraints are significantly harder to handle than cumulative (long-term) cost constraints since they require the policy to remain within the safe region at every time step, regardless of randomness. In practice, many SafeRL algorithms enforce such constraints through external mechanisms like safety layers, control barrier functions, or shielding.

2. **Probability of failure:** Here, for example, the constraint is defined as $\Pr(\text{eventual failure}) \leq \delta$. If one defines a cost $c(s, a)$ that is 1 upon entering a failure state and 0 otherwise, then $J_c(\pi)$ is “essentially” the (discounted) probability of failure. A constraint $J_c(\pi) \leq \delta$ limits failure probability. This can be handled in CMDP by that cost formulation Geibel and Wysotszki [2005].

Examples of Probability of Failure Constraints

Probability of failure constraints aim to limit the chance that an agent enters a catastrophic or irrecoverable state throughout its lifetime. As discussed, such constraints can be formalized by defining a cost function $c(s, a)$ which equals 1 when taking an action a in state s leads to a *failure state* (or belongs to a set of failure states), and 0 otherwise. The expected cumulative cost $J_c(\pi)$ under this formulation directly corresponds to the probability of failure. Below are typical scenarios where such constraints are relevant.

Spacecraft and Autonomous Vehicles — Safe Landing or Docking Probability Blackmore et al. [2010], Ono and Williams [2015, 2013], Chow et al. [2015]: In space missions or autonomous

landing scenarios, failure is often defined as crashing during landing or docking. One may enforce a constraint such as $\text{Pr}[\text{crash}] \leq \delta$, where δ is a small acceptable risk level. The cost function is defined as $c(s, a) = 1$ if (s, a) leads to a crash state.

Specifically, The paper Blackmore et al. [2010] is one of the earliest works in chance-constrained motion planning and is frequently cited in spacecraft and UAV planning and Ono and Williams [2015], is a classic reference on chance-constrained formulations for spacecraft landing and docking. The paper Ono and Williams [2013] directly deals with probability of failure constraints for spacecraft control.

Robotics — Falling or Tipping Over Berkenkamp et al. [2017], Wabersich and Zeilinger [2018], Berkenkamp and Schoellig [2015], Thananjeyan and et al. [2021]: In humanoid or legged robots, failure is typically associated with falling down. The agent is required to maintain $\text{Pr}[\text{fall}] \leq \delta$ to ensure physical integrity and task feasibility. In this case, any state classified as “fallen” is marked as a failure state, and $c(s, a) = 1$ if the next state is a fallen state. Specifically, Berkenkamp et al. [2017] is a classic paper that specifically addresses falling in legged robots and balance maintenance as a safety constraint. They model unsafe states (like falls) and ensure with high probability that they are avoided. The paper Wabersich and Zeilinger [2018] introduces a safety certification approach ensuring that robots do not enter dangerous states (including falls). It applies to both wheeled and legged robots. The earlier work Berkenkamp and Schoellig [2015] focuses on safe policy learning for balancing and preventing falls. It explicitly models unsafe states (falls) in the dynamics and safety set. Thananjeyan and et al. [2021] addresses falling probability directly in RL-based locomotion. It models $\text{Pr}[\text{fall}] \leq \delta$ type constraints explicitly for quadruped robots.

Healthcare — Patient Mortality or Critical Failure: In reinforcement learning for clinical decision-making (e.g., ICU treatment policies), a failure may be defined as the patient’s mortality or reaching a critical medical condition. The constraint $\text{Pr}[\text{critical_failure}] \leq \delta$ limits the treatment policy to maintain acceptable risk levels. Here, failure states correspond to medical emergencies.

In particular, Raghu et al. [2017], models ICU treatment as an MDP where mortality is treated as an absorbing failure state. While optimizing expected return, they explicitly consider trajectories leading to death. In 2018, Komorowski et al. [2018] proposed one of the first full-scale RL systems for ICU treatment. It models mortality as an outcome, and the trained policies aim to reduce the occurrence of death. Following this in Gottesman et al. [2019], proposed a foundational paper outlining safety and interpretability concerns in clinical RL. It explicitly discusses mortality and adverse outcomes as critical failure events. While not formalizing constraints as $\text{Pr}[\text{failure}] \leq \delta$, it motivates their necessity. In the same year, Jiang et al. [2019] explored the importance of constraint enforcement for safe clinical decision-making. It explicitly addresses risk of critical events in clinical policies.

Finance — Bankruptcy or Insolvency Events Jiang et al. [2019], Filar et al. [1995], Borkar and Jain [2014], Chow et al. [2015], Huang et al. [2018], Ravanbakhsh et al. [2019]: In financial portfolio management, the failure event could be the agent’s wealth dropping below a bankruptcy threshold. The probability of this event is often constrained by $\text{Pr}[\text{bankruptcy}] \leq \delta$ to limit risk exposure. The cost function is $c(s, a) = 1$ if wealth crosses the bankruptcy boundary.

The paper Filar et al. [1995] is one of the early papers introducing risk-sensitive portfolio optimization with Markov decision processes. It addresses ruin (bankruptcy) probabilities explicitly. In 2014 paper Borkar and Jain [2014], proposes to directly deals with probability of ruin (bankruptcy) in constrained MDPs. It proposes algorithms under constraints like $\text{Pr}[\text{bankruptcy}] \leq \text{Pr}[\text{bankruptcy}] \leq \delta$. The paper Huang et al. [2018] models percentile-based risk for financial RL tasks where falling below a wealth threshold triggers bankruptcy. While applied to cloud scheduling, Ravanbakhsh et al. [2019] demonstrates the same probability of ruin modeling, similar to financial insolvency constraints.

Manufacturing — Production System Breakdown: In industrial automation, a failure might occur when production machinery exceeds thermal, mechanical, or chemical safety limits

leading to breakdown. The probability of such system failure is constrained to be below a pre-specified threshold, e.g., $\Pr[\text{breakdown}] \leq \delta$.

Power Grids — Blackout Events: In power system control, blackouts (large-scale power failures) are often modeled as absorbing failure states. The system may enforce $\Pr[\text{blackout}] \leq \delta$ to reduce the chance of a cascading failure. Failure is usually caused by overloading, component failures, or instability.

In all these scenarios, $J_c(\pi)$ acts as the failure probability and CMDPs provide a natural framework for enforcing such probabilistic constraints.

3. **Risk measures:** Instead of expectation of cumulative cost, one could constrain a risk measure of the return (or cost). For example, constrain the variance of return below a threshold, or ensure $\text{CVaR}_\alpha(\text{cost}) \leq \delta$. Some works incorporate CVaR into RL as a way to ensure low probability of catastrophic outcomes Chow et al. [2015]. These constraints often do not fit the linear structure of CMDPs, but can be tackled with specialized algorithms.

Examples of Risk Measure Constraints

Risk measure constraints go beyond the expectation of cumulative cost and aim to control higher-order statistics or tail behavior of the cost distribution. These constraints are useful when we are concerned not only with average performance but also with rare but high-impact events. The most common risk measures in SafeRL include variance, Value-at-Risk (VaR), and Conditional Value-at-Risk (CVaR). Below are several examples from real-world applications.

Autonomous Driving — Variance-Constrained Driving Comfort: While avoiding collisions is a safety constraint, maintaining comfortable driving also involves controlling the variance of acceleration, jerk, or lane deviations. A variance constraint of the form $\text{Var}[\sum_t c(s_t, a_t)] \leq \delta$ can ensure that passenger discomfort due to aggressive or unstable maneuvers remains limited, reducing the risk of loss of control or accidents.

Specifically, in 2012, Tamar et al. [2012], introduced variance-constrained reinforcement learning where variance of return is explicitly controlled. It is applicable to driving scenarios when controlling variance of acceleration or jerk. Zhu et al. [2020] explicitly focuses on reducing control variability to improve smoothness and driving comfort. In 2021 Kiran et al. [2021] wrote a comprehensive survey that discusses driving comfort (acceleration, jerk minimization, smoothness) as key secondary objective in autonomous driving, and mentions various papers that incorporates constraints.

Finance — CVaR-Constrained Portfolio Optimization: In financial portfolio management, it is common to limit the Conditional Value-at-Risk (CVaR) of the portfolio’s return. A CVaR constraint $\text{CVaR}_\alpha[\text{loss}] \leq \delta$ ensures that the expected loss in the worst $\alpha\%$ of cases does not exceed a tolerable threshold. This is widely used to manage downside risk beyond what variance alone captures.

A seminal paper on optimization of conditional Value-at-Risk was by Rockafellar and Uryasev [2000] for portfolio problems. In 2014, Prashanth [2014] specifically focuses on CVaR-constrained MDPs with application to financial risk. This is the go-to reference in both optimization and financial risk management. In 2015, Chow et al. [2015], introduces CVaR-constrained RL applicable to portfolio optimization and other decision-making tasks. It provides methods to enforce CVaR constraints in sequential decision-making. In the same year Tamar et al. [2015], introduces policy gradient methods for risk-sensitive criteria including CVaR. It directly applies to portfolio optimization under CVaR constraints.

Healthcare — CVaR for Adverse Outcomes: In healthcare applications such as treatment planning or resource allocation, minimizing the expected number of adverse events may not be sufficient. A CVaR constraint on cumulative adverse events or side effects ensures that treatment policies control the likelihood of rare but severe negative outcomes.

Although general, the paper Prashanth and Ghavamzadeh [2016] is frequently cited in healthcare RL as it provides risk-sensitive methods including CVaR for controlling adverse events. Mattei and Gopalan [2019] specifically applies CVaR-based reinforcement learning to the problem of sepsis treatment, modeling adverse outcomes such as mortality and organ failure. This is one of the most direct applications of CVaR in healthcare. A position paper Gottesman et al. [2019] emphasizes that minimizing expected adverse outcomes is insufficient and recommended the use of risk measures (e.g., CVaR). Although it doesn't present an algorithm, it motivates CVaR as a necessary tool in treatment planning.

To summarize, CVaR is used in healthcare RL to limit the risk of rare but severe adverse events, to model tail risk (e.g., mortality, critical organ failure, side effects), and to design safe treatment policies under uncertainty.

Supply Chain Management — Risk-Averse Inventory Control: In supply chains, stockout events (inventory drops below zero) cause disruptions. Instead of just minimizing expected stockouts, a CVaR constraint on stockout penalties ensures that even in rare demand spikes, the risk of large cumulative stockouts is controlled.

One of the foundational works on risk-averse inventory control, discusses CVaR and other risk measures Chen et al. [2014]. It models stockouts as undesirable events and controls the risk via dynamic programming. Widely used as a textbook, Shapiro et al. [2014] includes detailed treatment of CVaR in supply chain optimization. It explains how risk measures such as CVaR can control stockouts and demand uncertainties. While not supply chain specific, Chow et al. [2015] is often cited in supply chain literature for inventory control under CVaR constraints. Its techniques directly apply when modeling stockouts as risky events. A highly cited paper is Bertsimas and Thiele [2006] showing how robust optimization (a precursor to CVaR-type models) controls stockout risks. Provides insight into handling demand uncertainty and stockout penalties.

Robotics — CVaR-Constrained Trajectory Optimization: For autonomous robots navigating uncertain environments, one may use CVaR constraints on cumulative collision risk or energy consumption. This ensures that the robot does not just minimize average risk but is also robust against worst-case environmental uncertainties or adversarial perturbations.

The paper Majumdar and Pavone [2020] directly studies CVaR-constrained trajectory optimization for robots under uncertainty. Formulates trajectory optimization problems where collision risk is controlled using CVaR. The paper Zhang et al. [2020] focuses on risk-averse path planning under environmental uncertainty using CVaR. Provides algorithms and examples for safe robot navigation with collision risk control. The paper Singh et al. [2018] considers autonomous driving (viewed as mobile robotics) with risk-sensitive trajectory optimization. It uses CVaR to constrain collision probabilities and improve robustness to adversarial perturbations.

Power Systems — Risk-Sensitive Stability Control: In power grid operations, rather than just minimizing expected frequency deviations or power outages, operators may use CVaR or variance constraints to ensure that the probability of large-scale instabilities remains acceptably low, accounting for rare but impactful demand or supply fluctuations.

The paper Bitar and Low [2012] addresses reliability and demand uncertainties in power systems with a risk-sensitive approach. It Models constraints on load shedding and supply-demand balancing. The paper Dall’Anese et al. [2015] directly introduces chance-constrained optimization for voltage stability and power flow, which is equivalent to controlling the probability of instability; CVaR and probability bounds are discussed. In 2012, Jiang et al. [2012] introduces risk-constrained OPF formulations using CVaR and chance constraints. It ensures that the probability of voltage violations and instabilities is below a prescribed risk level. A year earlier in 2011, Zhang et al. [2011] models the variance and tail risk of power system instability due to fluctuating wind generation. While **not** CVaR directly, it motivates variance and higher-order moment-based risk constraints.

Power systems use variance, probabilistic, and CVaR constraints to: Avoid rare but catastrophic blackouts, to maintain voltage and frequency within safe margins, and to ensure reliability under demand and renewable generation uncertainty.

Risk measures provide a flexible modeling tool for specifying safety, robustness, and fairness. However, incorporating them often requires non-standard methods such as CVaR-optimized policy gradients, distributional reinforcement learning, or scenario-based optimization, as these constraints typically violate the linear structure required for classic CMDP formulations.

4. **Multi-objective viewpoint:** SafeRL can be seen as a multi-objective optimization where one objective is reward and others are (negative) costs. The constraint formulation picks one point on the Pareto frontier by treating costs as hard constraints. Alternatively, one could combine reward and costs into a single scalar reward via weighted sum (penalty method), but that requires tuning weights and does not guarantee constraint satisfaction Achiam et al. [2017]. Constrained formulation cleanly separates objectives and safety.

Examples of Multi-Objective Viewpoint in Safe Reinforcement Learning

In many real-world applications, agents must simultaneously optimize multiple objectives that may conflict. Typically, SafeRL is modeled as a multi-objective problem where one objective is the primary reward, while others are safety-related costs. The CMDP formulation addresses this by enforcing costs as hard constraints, selecting a specific point on the Pareto frontier. Alternatively, some works use a scalarization (penalty) method by combining reward and costs into a single objective. Below are common examples illustrating the multi-objective viewpoint.

Robotics — Speed vs. Safety Trade-off Achiam et al. [2017], Berkenkamp et al. [2017], Chow et al. [2018a]: A mobile robot navigating in an environment may aim to maximize the reward associated with reaching the goal quickly. However, it also needs to minimize the probability of collisions and energy consumption. Here, speed contributes positively to the reward, while collisions and energy usage are treated as negative costs. The CMDP formulation could enforce a maximum acceptable collision rate and energy budget, leading to an explicit safety-performance trade-off.

Autonomous Driving — Travel Time vs. Accident Risk Dalal et al. [2018], Shalev-Shwartz et al. [2017], Zheng and Gu [2024]: In autonomous driving, agents aim to minimize the expected travel time while simultaneously ensuring a low probability of accidents. The agent faces a trade-off between driving faster (leading to higher reward) and maintaining safe distances or reduced speeds to avoid collisions (cost). The Pareto frontier consists of policies ranging from conservative (low accident risk, long travel time) to aggressive (low travel time, high accident risk). SafeRL selects a policy on this frontier according to the safety constraint.

Energy Systems — Power Supply vs. Cost and Reliability Bitar and Low [2012], Dall’Anese et al. [2015]: In power grid management, the agent may aim to optimize electricity production to meet demand (reward) while minimizing costs associated with fuel consumption and the risk of violating reliability standards (costs). This problem naturally involves multiple objectives: maximizing supply quality and minimizing operational risks and costs.

Healthcare — Treatment Success vs. Adverse Effects Gottesman et al. [2019], Raghu et al. [2017]: In medical decision-making, an RL agent may need to maximize treatment efficacy while minimizing adverse effects or treatment toxicity. For example, maximizing patient recovery speed could conflict with the need to limit drug dosage to avoid harmful side effects. A CMDP constraint could limit the expected cumulative adverse effects to a tolerable threshold, enforcing safety.

Manufacturing — Production Efficiency vs. Maintenance Costs Chung et al. [2020], Khouja [2003]: In automated manufacturing, increasing production speed or output (reward) may

result in higher machine wear and maintenance costs (costs). A CMDP-based SafeRL framework may impose a constraint on expected maintenance cost or machine degradation, forcing the agent to balance throughput and longevity.

Drone Swarms — Task Completion vs. Communication Load Gu et al. [2023], Yang et al. [2020b]: In multi-drone systems, agents may wish to maximize task completion rates (reward) while minimizing communication overhead (cost). Communication constraints can act as safety constraints in environments with bandwidth limitations or interference risks.

In all these cases, treating costs as hard constraints via CMDPs gives a systematic way to trade off reward and cost by directly selecting a feasible point on the Pareto frontier. In contrast, using a scalarization approach (reward minus weighted costs) can lead to policies that violate constraints unless the weights are carefully chosen and tuned.

5. **Temporal logic specifications** Alshiekh et al. [2018a], ElSayed-Aly et al. [2021], Turchetta et al. [2016], Wabersich and Zeilinger [2018]: In some safety-critical settings, the safety requirement is given as a formal temporal logic formula (e.g., “always avoid region X unless Y happens”). Such logic specifications can be converted to automata and then to reward/cost functions or shields that enforce them Alshiekh et al. [2018a], ElSayed-Aly et al. [2021]. While not a traditional CMDP constraint, they can often be incorporated by extending the state space to include automaton states representing the satisfaction of the formula. Specifically, in Alshiekh et al. [2018a], Alshiekh et al. (2018) introduced safe reinforcement learning via shielding; they used LTL (Linear Temporal Logic) specifications to construct shields for RL agents. In Wabersich and Zeilinger (2018) Wabersich and Zeilinger [2018], linear model predictive safety certification for learning-based control was employed. Although it focused on model predictive safety, their framework is capable of incorporating logic-based safety constraints. The paper ElSayed-Aly et al. [2021] extends shield-based safe RL to the multi-agent setting using temporal logic specifications. The paper Turchetta et al. [2016] while focused on safe exploration, their work shows how formal safety specifications can be integrated into exploration guarantees. One of the older but influential paper Sadigh et al. [2016] shows how specifications from temporal logic can shape safe planning.

Throughout this survey, we largely assume the standard expected cumulative cost constraints unless stated otherwise. This assumption covers many practical cases (like average constraint violation rate, or total resource consumption) and has well-developed theoretical tools. When discussing specific algorithms, we will note what type of constraint they handle (most often, it is expected cost).

3.4 Theoretical Results

We highlight a few key theoretical results for CMDPs relevant to SafeRL:

Theorem 3.1 (Optimal Policy for CMDP Altman [1999]). *For a finite CMDP with bounded rewards and costs, there exists an optimal policy (π^*, λ^*) that attains the maximum in (1) (and corresponding optimal dual variables). Moreover, there exists an optimal policy that is stationary (time-independent) and can be chosen to be deterministic with respect to actions at all but possibly a measure-zero set of states. In practice, optimal policies may randomize between a small number of deterministic policies if needed to exactly satisfy constraints.*

In short, one does not need complex history-dependent or non-Markovian policies to solve CMDPs optimally; memoryless policies suffice, simplifying the search space for algorithms.

Theorem 3.2 (Lagrange Duality Altman [1999]). *Under mild regularity conditions (e.g., finite state/action or convexity in policy space), The strong duality holds for the CMDP problem. That is,*

$$\min_{\lambda \geq 0} \max_{\pi} \mathcal{L}(\pi, \lambda) = \max_{\pi} \min_{\lambda \geq 0} L(\pi, \lambda),$$

and solving the dual yields the primal optimum. The optimal dual variables λ^ provide valuable information: if $\lambda_i^* > 0$, then the i -th constraint is active (tight) at the optimum; if $\lambda_i^* = 0$, the optimum policy naturally satisfies i -th constraint with some slack.*

This theorem justifies many SafeRL approaches that focus on solving the dual via gradient methods on λ while finding optimal policies for a given λ using RL.

Proposition 3.3 (Policy Gradient for Constrained Objectives). *If the policy π_θ is parameterized by θ (e.g., a neural network), one can derive gradients for the constrained problem. For instance, using the Lagrangian, the gradient of $\mathcal{L}(\pi_\theta, \lambda)$ with respect to θ is*

$$\nabla_\theta \mathcal{L} = \nabla_\theta J(\pi_\theta) - \sum_i \lambda_i \nabla_\theta J_c^{(i)}(\pi_\theta).$$

This leads to constrained policy gradient algorithms, where θ is updated in the direction of $\nabla_\theta \mathcal{L}$ and λ is updated in the direction of $\nabla_\lambda \mathcal{L} = d_i - J_c^{(i)}(\pi_\theta)$. Many actor-critic style SafeRL methods employ this simultaneous gradient update (a form of primal-dual gradient descent) Chow et al. [2018b].

Safe exploration and probably safe learning: A distinction in SafeRL theory is between methods that guarantee *safety during learning* vs. only *at convergence*. Most theoretical results (like the ones above) ensure that the final learned policy can satisfy constraints. Ensuring that intermediate policies (during training) also satisfy constraints is much harder. Constrained policy optimization approaches (Section 4) aim to maintain safety at each iteration by conservative updates Achiam et al. [2017]. Another line of work uses PAC-style analysis or high-probability bounds to derive exploration strategies that with high probability never violate constraints beyond a tolerance Turchetta et al. [2016], Berkenkamp et al. [2017]. These often rely on optimistic models or Lyapunov functions to formally verify safe regions of state-space the agent can explore. Though we do not delve into detailed proofs, we note that providing safety guarantees during learning typically requires additional assumptions (such as mild system dynamics, or an initial safe policy to bootstrap from). Having established the CMDP framework and theoretical background, we now move on to discuss concrete algorithms and methods developed for SafeRL, both in the single-agent case (Section 4) and multi-agent extensions (Section 5).

4 State-of-the-Art Methods in SafeRL and SafeMARL

In this section, we survey major methods and algorithms in Safe Reinforcement Learning, covering both single-agent SafeRL in CMDP settings and extensions to multi-agent SafeMARL. We organize the discussion by methodological categories, explaining how each approach incorporates safety and highlighting key algorithms. For each category, we provide examples of state-of-the-art techniques and cite representative works.

4.1 Lagrangian-based Policy Optimization

One broad class of SafeRL algorithms uses the primal-dual (Lagrangian) approach discussed earlier to enforce constraints. The idea is to transform the constrained problem into a sequence of unconstrained problems with adjusted rewards.

Lagrangian Actor-Critic: In this approach, one augments the standard RL loss with penalty terms for constraint costs. For example, one can define a penalized reward $r_\lambda(s, a) = r(s, a) - \lambda c(s, a)$ (eqn (11) in Tessler et al. [2019]) for a single-constraint problem, where λ is treated as a learnable parameter. An actor-critic algorithm (Algorithm 1 in Tessler et al. [2019]) can then be used: The *actor* (policy π_θ) is updated with respect to the penalized objective $J_{\text{pen}}(\pi) = J(\pi) - \lambda J_c(\pi)$, using policy gradient or other optimization. The *critic*(s) estimate both the value of the reward and the cost (often one critic for $V^\pi(s)$ and one for $V_c^\pi(s)$). The Lagrange multiplier λ is updated by gradient ascent on the constraint satisfaction term, e.g. $\lambda \leftarrow \lambda + \beta(J_c(\pi) - d)$. This simple scheme is often called the *Lagrange method* or *reward shaping method* in safe RL. It was used in early safe deep RL implementations (e.g., Tessler et al. [2019] for safe DQN with constraints, and policy-gradient variants, i.e., Trust Region Policy Optimization and Proximal Policy Optimization in Ray et al. [2019]). While straightforward, a drawback is that the penalty coefficient λ can be hard to tune (“Our baseline results for constrained RL indicate a need for stronger and/or better-tuned algorithms to succeed on Safety Gym environments” as quoted in Ray et al. [2019]) and the method does not guarantee strict constraint satisfaction until convergence. The agent might violate constraints during learning if λ is not large enough, or conversely, learn too slowly if λ is too large initially.

Projected Lagrangian (Constrained Policy Optimization): Achiam et al. Achiam et al. [2017] introduced **Constrained Policy Optimization (CPO)**, a landmark algorithm that improves upon the basic Lagrangian method by ensuring each policy update is safe. CPO is built on trust-region policy optimization: At each iteration, it solves a local constrained optimization: maximize policy improvement subject to a constraint that the cost does not increase beyond a small tolerance. This is done by a quadratic approximation of the objective and a linear approximation of the constraints (using policy gradient and cost gradient), then solving a convex subproblem. If the proposed update violates the constraint (predicted cost increase too high), CPO backtracks or projects the policy update to the nearest feasible update. CPO provides theoretical guarantees of *near-constraint satisfaction at each iteration*: essentially, it never overshoots the constraint by more than a certain second-order error term, keeping training safe. CPO demonstrated that one can train neural network policies for control tasks while maintaining safety throughout training Achiam et al. [2017]. It was the first general-purpose safe RL algorithm with such guarantees. However, CPO is more complex and computationally heavier than standard policy gradient (due to solving the constrained optimization subproblem each step). It also requires a reliable estimation of the cost value and cost advantage, which can be challenging. Many subsequent works have built on or modified CPO: **PCPO (Projection-based CPO):** an algorithm that explicitly projects the policy gradient to the feasible set defined by constraint gradients Yang et al. [2020a]. It is a simplification that avoids solving a quadratic program but still aims to keep updates safe by geometric projection.

TRPO-Lagrangian: A simpler baseline where one applies a trust-region update on the penalized objective $J - \lambda J_c$ instead of solving a constrained QP. This does not guarantee strict feasibility but often empirically manages constraint violations by proper λ adaptation. OpenAI’s Safety Gym benchmark release OpenAI [2019], Ray et al. [2019] used such baselines¹.

Actor-Critic with Lyapunov: Chow et al. Chow et al. [2018a] proposed using a Lyapunov function (a monotonic function of the cost-to-go) to derive a safe update rule. They ensure the new policy does not increase a Lyapunov function, which in turn guarantees the constraint remains satisfied. This can be seen as another form of trust-region or projection method specialized using Lyapunov theory.

Off-policy and Model-based extensions: While most policy optimization methods are on-policy, there have been adaptations to off-policy learning: *Safe DDPG or TD3*: by incorporating

¹<https://github.com/openai/safety-starter-agents>

a cost critic and Lagrange multiplier, one can train deterministic policies (as in DDPG) with a constraint. For example, a constrained variant of TD3 (Twin Delayed Deep Deterministic Policy GradientFujimoto et al. [2018]²) was proposed by Kostrikov et al. [2021], Lyu and Liu [2021].

Model-based SafeRL: Berkenkamp *et al.* Berkenkamp et al. [2017], Berkenkamp and Schoellig [2015], Berkenkamp et al. [2016] used Gaussian process models of the dynamics to ensure safety. They construct a stabilizing controller (via control theory) that acts as a baseline policy and only allow the learning agent to explore if it can certify (using a Lyapunov condition) that the new policy is safe. While not directly a CMDP approach, this provides an alternative angle: blending traditional control safety with RL exploration.

4.2 Safety Shields and Action Correction

Another category of SafeRL methods focuses on safe exploration: how to prevent an agent from ever taking an action that could lead to catastrophe. These methods act as a layer on top of any standard RL algorithm, modifying or filtering its actions:

- **Safety Shield / Filter:** A mechanism that monitors the agent’s chosen action and overrides it if it is deemed unsafe. The override might be a safe default action or the closest safe action. Dalal *et al.* Dalal et al. [2018] introduced a *safety layer* that solves a quadratic program (QP) in continuous action spaces to minimally adjust the action such that predicted next state stays within safety bounds. This method guaranteed zero constraint violations during training on those tasks. However, it requires a model (or learned model) to predict constraint violations.
- **Shielding via formal methods:** Alshiekh *et al.* Alshiekh et al. [2018b] and later ElSayed-Aly *et al.* ElSayed-Aly et al. [2021] (extended to multiagents) use formal verification and temporal logic to build shields. The idea is to pre-compute a set of forbidden state-action pairs using model checking of an abstract model, or to synthesize a runtime observer from a formal specification. The shield then blocks any action that would lead into a bad state (violating the LTL safety specification) in finite steps. In multi-agent settings, as ElSayed-Aly et al. [2021] shows, one can have a centralized shield watching over joint actions or distributed shields for each agent.
- **Human or Oracle intervention:** In practical scenarios, one may employ a human overseer or a safety oracle to intervene when the agent is about to do something unsafe. While not a scalable solution for all time, during training it can prevent disasters. Safe RL with human intervention was studied in Saunders et al. [2017] where a human can cancel dangerous actions, and the agent is penalized for those. Over time the agent learns to avoid actions that would have been blocked.

Shielding approaches have the advantage of hard safety (no violations in theory), but they often rely on having additional knowledge: either a dynamics model, a predefined safe set, or an external supervisor. They also may introduce performance bias (the agent might become too conservative if the shield is not carefully designed, since it never experiences certain parts of state space). Combining shielding with CMDP-based learning is an interesting direction: one can use shielding in early training and gradually lift it as the agent’s own policy becomes safe with learned constraints.

4.3 Risk-Sensitive and Distributional Methods

Although our focus is on constraint-based SafeRL, a brief mention of risk-sensitive RL is warranted as an alternative approach: In risk-sensitive RL, instead of constraints, the optimization criterion itself is altered to account for risk. For example, one might maximize $U^{-1}(E[U(\sum r)])$ where U is a concave utility (exponential utility gives risk-aversion) or maximize $\text{CVaR}_\alpha(\sum r)$ of return at some

²<https://spinningup.openai.com/en/latest/algorithms/td3.html>

confidence level α . Tamar *et al.* Tamar et al. [2015] and others have developed policy gradient methods for CVaR. These effectively try to ensure with high probability the return is above some level, which is conceptually similar to constraints on probabilities of bad events. **Distributional RL** (as popularized by Bellemare et al. [2017]) learns the full distribution of returns. One can combine distributional RL with safety by focusing on the lower tail of the return distribution to ensure it is above some threshold. This is another way to encode safety without explicit constraints. Risk-sensitive methods can sometimes be converted into CMDP style constraints. For instance, requiring $\text{CVaR}(\text{cost}) \leq d$ is a constraint on a specific risk measure of cost. Solving such constraints often introduces auxiliary variables or uses sample-based approximations. While we do not detail these methods here, they are part of the broader SafeRL toolbox.

4.4 Safe Multi-Agent Reinforcement Learning (SafeMARL)

SafeMARL extends the ideas above to multi-agent systems Albrecht et al. [2024], Weiss [1999], Wooldridge [2009], Shoham and Leyton-Brown [2008], Weiss [1996]. We consider environments with N agents, indexed by $i \in 1, \dots, N$. A convenient formal model is a **constrained Markov game**, defined by $(\mathcal{S}, \mathcal{A}_i, P, r_i, c_i^{(j)}, \gamma)$. Here each agent i chooses an action $a_i \in \mathcal{A}_i$, forming a joint action $\mathbf{a} = (a_1, \dots, a_N)$ that causes state transitions via $P(s'|s, \mathbf{a})$. Each agent can receive an individual reward $r_i(s, \mathbf{a})$ and has possibly its own set of cost functions $c_i^{(j)}(s, \mathbf{a})$ for $j = 1 \dots m_i$. SafeMARL scenarios can be cooperative, competitive, or mixed

- In fully **cooperative SafeMARL**, all agents share a common reward (or their rewards are aligned) and typically the safety constraints are also shared or at least all agents are interested in satisfying all constraints. For example, a team of robots might have a joint goal (maximize sum of rewards) and constraints like “no collisions among any robots” which is a global safety constraint.
- In **competitive or general-sum SafeMARL**, each agent has its own reward to maximize, and constraints might be individual (each agent has its own safety requirement) or shared (environment-level safety that everyone needs to uphold, like traffic rules). The solution concept might be a safe equilibrium Ganzfried [2022] (e.g., a Nash equilibrium that respects constraints, a related work Everitt et al. [2019] considers safe strategies for players) rather than a single policy optimization. Most existing work in SafeMARL addresses cooperative settings, since even standard MARL is most tractable in either fully cooperative (centralized training for a team) or fully competitive (two-player zero-sum) cases.

We highlight a few key approaches:

- **Centralized Training with Global Constraints:** A straightforward extension of single-agent SafeRL to multi-agent cooperative tasks is to treat the entire multi-agent system as one big agent with a joint action Albrecht et al. [2024]. One can then apply CMDP methods on the joint system. For example, one can define a joint policy $\pi(\mathbf{a}|s)$ and a global cost $c_{\text{global}}(s, \mathbf{a})$ that encodes any violation by any agent. Then apply CPO or Lagrange methods on this joint policy. This was essentially the approach in the MACPO algorithm³ by Gu et al. [2023]: they derived a multi-agent version of the CPO update (ensuring monotonic improvement in team reward and satisfaction of safety constraints). In practice, they implemented MACPO with two variants: one using a centralized critic (accessible during training) that estimates global reward and cost, and another using a factorized approach (MAPPO-Lagrangian, See Lemma 1: Multiagent advantage decompositon in Gu et al. [2023]) which is simpler and uses decentralized advantage estimates with a Lagrange penalty for costs. The challenge with centralized approaches is the scalability: the joint action space grows exponentially with number of agents, and a centralized policy might be impractical for many agents. It also requires a central controller during training (and possibly execution) that knows all agent’s states, which might not be available in all applications.

³<https://github.com/chauncygu/Multi-Agent-Constrained-Policy-Optimisation>

- **Decentralized Safe Learning with Coordination:** An important research direction is how to achieve safe multi-agent learning without relying on a central entity or a global state accessible to all. Recent work by Zhang *et al.* Zhang et al. [2024] introduced a scalable constrained policy optimization where each agent optimizes a localized objective that approximates the global safety. They use the concept of κ -hop neighborhood (each agent coordinates with others within κ hops in a communication graph) to truncate the dependence on far-away agents. They proved that if each agent optimizes a local policy with these truncated safety constraints and updates sequentially, the overall system still improves reward and satisfies constraints. The resulting algorithm (Scalable MAPPO-Lagrangian) shows promising results on large multi-agent environments, demonstrating that strict centralization is not always necessary for SafeMARL. Another method for decentralization is to factor the safety constraints: ElSayed-Aly et al. [2021] did this via shields for subsets of agents. In general, one can attempt to decompose a global constraint into local constraints for each agent. For example, a global cost $c_{\text{global}}(s, \mathbf{a})$ might be decomposed as $c_{\text{global}} = \sum_i c_i(s, a_i)$ if the unsafe events are localized per agent. Then each agent could constrain its own c_i . However, not all safety constraints are additively separable; many (like collision avoidance) are inherently about joint configurations. This remains a hard problem: designing local reward/cost structures whose alignment with global safety yields provable guarantees.
- **Multi-agent Credit Assignment for Safety:** In multi-agent RL, credit assignment (determining each agent’s contribution to global reward) is crucial. Similarly, for safety, one might need to assign “blame” or responsibility to individual agents for a safety violation. Approaches like difference rewards or shaped team rewards can be used to ensure each agent gets feedback about how its actions affected the global outcome. For SafeRL, one could design each agent’s cost signal such that it corresponds to the marginal increase in global risk due to that agent. Some initial works have considered approaches where each agent considering the safety constraints of others Gu et al. [2023], though a general solution is open research (we outline this as a problem later).
- **Safe Equilibria and Non-Cooperative Agents:** For competitive settings, one could consider each agent solving its own CMDP subject to safety constraints, leading to a game where each agent’s strategy must satisfy its own constraints. The concept of a *constrained Nash equilibrium* arises: a profile of policies π_1, \dots, π_N such that no agent can improve its reward without violating constraints given the other’s policies. Algorithms to compute such equilibria are not well-developed; this might involve ideas from game theory (like best response dynamics with constraints or Lagrangian for each agent). One example in literature is safe multi-agent learning via Stackelberg games: one agent (leader) accounts for the follower’s response. Zheng and Gu [2024] apply a bilevel optimization (Stackelberg) to model an autonomous driving scenario with safety, effectively solving a two-agent safe RL where the vehicles plans with knowledge of the other’s constraints (for example, in road intersection environments). This is a rich area for future investigation.
- **Benchmarking SafeMARL:** The progress in SafeMARL has been accelerated by the introduction of benchmarks. Gu *et al.* Gu et al. [2023] provided *Safe Multi-Agent MuJoCo*⁴, *Safe MARobosuite*, and *Safe MA-IsaacGym*, which are multi-robot simulation tasks with safety constraints (like torque limits or collision constraints). These environments allow systematic evaluation of SafeMARL algorithms in settings requiring coordination. Similarly, for single-agent SafeRL, OpenAI’s *Safety Gym*⁵ Ray et al. [2019] introduced a suite of continuous control tasks with hazards and constraints, which has become a standard testbed.

In summary, state-of-the-art SafeRL methods range from modified policy gradient algorithms (with theoretical guarantees like CPO) to pragmatic penalty-based methods, model-based safe

⁴<https://github.com/chauncygu/Safe-Multi-Agent-Mujoco>

⁵<https://github.com/openai/safety-gym>

Method/Algorithm	Description and Key Features
Lagrangian actor-critic Tessler et al. [2019]	Add constraint cost as penalty to reward; update λ online. Simple but may violate constraints before convergence.
Constrained Policy Optimization (CPO) Achiam et al. [2017]	Trust-region policy updates with theoretical guarantee of near-constraint satisfaction each iteration. Uses second-order approximations to ensure safety.
Lyapunov-based Policy Optimization Chow et al. [2018a]	Uses a Lyapunov function (cost critic) to constrain updates. Guarantees decrease in an upper bound of cost.
Reward Constrained DQN Tessler et al. [2019]	DQN with a reward penalty for constraint, ensuring discrete actions respect cost limit in expectation.
Safe DDPG/TD3 (Lagrangian)	Extends continuous control off-policy algorithms with cost critics and Lagrange multipliers for constraints.
Safe Model-Based RL Berkenkamp et al. [2017]	Uses model uncertainty estimates and stability analysis to allow only proven-safe explorations. Ensures no violations under certain dynamics assumptions.
Safety Layer (action shield) Dalal et al. [2018]	A differentiable layer that projects chosen actions to the nearest safe action by solving a QP. Guarantees zero immediate violations given local dynamics linearization.
Shielding (LTL) ElSayed-Aly et al. [2021]	Pre-compute shields from formal specifications; filter multi-agent joint actions to avoid unsafe outcomes. Achieves provable safety with respect to spec.
Multi-Agent CPO (MACPO) Gu et al. [2023]	Extension of CPO for multi-agent teams. Centralized training, uses a joint policy or coordinated update. Demonstrated on multi-robot tasks.
Scalable Decentralized Safe MARL Zhang et al. [2024]	Each agent optimizes a local surrogate constrained problem using truncated observation of others. Achieves near-centralized performance with better scalability.
Safe MARL via Bilevel (Stackelberg) Zheng and Gu [2024]	Models one agent as leader, others as followers in a game with safety constraints. Solves via bilevel optimization to account for interactive safety.
Safe MARL with Shielding ElSayed-Aly et al. [2021]	Combines MARL with runtime shielding (central or factored) to ensure no unsafe joint actions are taken during learning.

Table 1: Representative SafeRL (single-agent) and SafeMARL (multi-agent) methods.

exploration, and safety layers, whereas SafeMARL is exploring centralized vs. decentralized learning, coordination mechanisms, and safe policy equilibrium concepts. Table 1 provides a high-level summary of key selected algorithms in SafeRL and SafeMARL.

5 Open Research Challenges and Future Directions

While significant progress has been made in SafeRL and SafeMARL, many challenges remain open. In this section, we present five research problems that, if solved, would substantially advance the field. Three of these pertain specifically to SafeMARL, reflecting the newer nature of multi-agent safety. For each problem, we describe the motivation, outline possible approaches (steps toward a solution), and reference relevant prior work to build upon.

Most SafeRL algorithms guarantee safety in expectation or asymptotically, but they often allow some violations during learning (especially early on). In high-stakes applications, even a single catastrophic failure is unacceptable. The research challenge is to design RL methods that ensure *zero (or provably bounded) constraint violations throughout the entire training process*, without relying on a human in the loop.

Ensuring no violations typically requires either very conservative exploration or prior knowledge (dynamics models, safe baseline policy). Too conservative an approach can severely slow down learning. Balancing caution with exploration is tricky, as overly restrictive safety can trap the

policy in a local optima (not exploring better solutions).

The concept of never violating constraints relates to **safe exploration**. Moldovan and Abbeel (2012) and others studied conditions for “safe policy learning” where certain states are absorbing traps (unsafe) and should be avoided forever. Approach like *learning with a safety critic* Thananjeyan and et al. [2021] or *mentor-assisted exploration* Saunders et al. [2018], Zhou and Li [2018], Curi et al. [2020] have been tried. However, a general solution remains elusive, especially for high-dimensional continuous tasks. Solving this problem would likely require combining learning with elements of control theory or formal methods to get the needed guarantees.

Many real-world problems are partially observable (POMDPs) – the agent does not have full knowledge of the true state relevant to safety. Examples: a robot with limited sensors, or an autonomous car that cannot see around corners. In such cases, ensuring safety is harder because the agent might inadvertently take an unsafe action due to missing information. The challenge is to design SafeRL algorithms that operate under uncertainty/partial observability and still guarantee safety.

In a POMDP, the agent typically maintains a belief (distribution over states). Constraints might need to be satisfied with respect to the true state (which is unknown). For instance, we might require that *for all possible true states consistent with the agent’s observations, the safety constraint holds*. This is a very strict condition and can be overly conservative. Alternatively, one might demand a high probability of safety given the belief.

There is work on **POMDPs with chance constraints**, where constraints must hold with a certain probability. Techniques often convert these into augmented state MDPs by including some memory or using scenario optimization. Another related concept is **belief shielding**: e.g., using human feedback to avoid ambiguous unsafe states. Solving safe RL in POMDPs could connect to robust control in partially observed systems (like robust Model Predictive Control with chance constraints). This problem remains largely open; progress would benefit fields like autonomous systems operating with imperfect sensors.

In many multi-agent applications, each agent has only partial, local observations (e.g., each car in traffic sees only nearby cars). A central authority that monitors and enforces safety for all agents may not exist. The challenge is to achieve safe multi-agent learning in a *fully decentralized* way: each agent makes decisions based on its local view and (optionally) limited communication, and together their behaviors ensure global safety constraints are respected.

Global safety constraints often involve the joint state of multiple agents (e.g., distance between any two drones must exceed a threshold to avoid collision). No single agent can evaluate the global constraint alone. If communication is limited (bandwidth or range), agents might not know the actions or states of others in time to react safely. Moreover, learning is now on a game (or team) level, complicated by non-stationarity (each agent’s environment is affected by others learning simultaneously).

Decentralized MARL has been studied (e.g., independent learners, mean-field MARL), but safety adds extra difficulty. Zhang et al. [2024] is one of few works aiming at decentralized Safe-MARL. Also relevant are **distributed constrained optimization** in control theory where multiple controllers ensure a global constraint (like distributed frequency control in power grids ensuring safety constraints on voltage). Techniques from **graphical games** or **networked control systems** could be applied. Success in this problem would directly impact fields like distributed robotics and network safety (e.g., ensuring no network congestion collapse via decentralized RL controllers).

SafeRL research has mostly focused on a single agent or cooperative teams. However, in the real world, multiple independent agents (e.g., companies trading stocks with safety limits, or autonomous cars from different manufacturers) may not share a common goal. They might even be adversarial. Each has safety constraints (like not going bankrupt, or not crashing) but they also have competing objectives. The challenge is to extend SafeRL to **general-sum or competitive environments**, finding appropriate equilibrium notions and algorithms to compute them.

In competitive multi-agent scenarios, one cannot simply optimize a joint objective. Methods like CPO do not directly apply, because improving one agent’s reward might hurt another’s. We need an equilibrium concept like *constrained Nash equilibrium* or *constrained correlated equilib-*

rium. Another issue is that safety for one agent might depend on the behavior of others. If others act recklessly, an agent might be unable to guarantee its safety without overly sacrificing reward (or might need to assume worst-case opponents).

Constrained game equilibria have been studied in economics (e.g., Nash equilibria with budget constraints). In RL, one related field is **Mean-Field Games with constraints**, but results are sparse. Another is **Multi-agent reinforcement learning for traffic**, where multiple self-interested cars must avoid collisions (a safety constraint) – some works use hand-crafted rules or potential fields, but learning such behavior while each optimizes their own objective is largely open. If this problem is solved, it could define how autonomous systems from different stakeholders safely coexist (think of air traffic control but without a central controller—planes negotiating to avoid collisions while meeting their own goals).

Consider multi-agent systems that operate over long time scales where the environment or the set of agents may change. For example, a fleet of autonomous vehicles might encounter new types of vehicles or changing traffic rules; or a robotic factory might add/remove robots over time. We need SafeMARL algorithms that can **adapt to non-stationarity** in the environment or agent population, while preserving safety. This includes scenarios like agents entering or leaving, changes in the dynamics, or even adversarial perturbations.

Non-stationarity breaks the assumptions of convergence for most RL algorithms. SafeRL adds another layer: after training, if something changes and the policy is no longer safe, the agent must detect and correct this quickly (ideally without catastrophic failure during the transition). Multi-agent adds complexity because one agent’s non-stationarity (learning or adapting) is another agent’s non-stationary environment.

Non-stationary RL is a growing area (sometimes framed as lifelong learning or non-stationary bandits). SafeRL in non-stationary settings has seen little study. One relevant angle is **robust safe RL**: algorithms that ensure safety under model perturbations (e.g., Chen et al. [2021], Yang et al. [2021] considered adversarial changes in cost function within limits). Another is **meta-learning safety**: a recent work by Grbic and Risi [2020] attempted to meta-learn a safety critic for quickly evaluating new scenarios. Achieving continual safe learning could pave the way for real-world deployment where conditions are never static.

Summary of Research Directions

The problems outlined above are interconnected. For example, solving safe exploration (Problem 1) will likely benefit safe adaptation (Problem 5); and advances in decentralized safe learning (Problem 3) will be crucial for tackling competitive safe learning (Problem 4) where a central authority is absent. Each problem requires a blend of techniques—RL algorithms, optimization theory, control theory, and even insights from economics or game theory. Crucially, addressing these problems will move SafeRL from a laboratory curiosity to a dependable component of autonomous systems. We expect that success in these areas will result in publishable work at top venues (NeurIPS, ICML, ICRA, etc.), given the importance and difficulty of ensuring safety in learning systems. By formulating them here, we hope to encourage more researchers to contribute to these challenges.

6 Conclusion

Safe Reinforcement Learning is a vital area of research for deploying learning agents in real-world environments where failures are costly or dangerous. In this survey, we provided a detailed overview of SafeRL with a focus on the CMDP framework for incorporating constraints. We reviewed theoretical foundations including CMDP definitions, Lagrangian duality, and solution methods like linear programming and policy gradient for constrained problems. Building on this foundation, we discussed state-of-the-art SafeRL algorithms such as Constrained Policy Optimization, Lagrange multiplier methods, safe exploration via shielding, and how these have been extended to multi-agent settings. Our survey highlights that:

- SafeRL is inherently a cross-disciplinary field, drawing from machine learning, optimal control, and formal methods. The CMDP formulation provides a unifying language for many approaches.
- There is a rich toolbox of algorithms for single-agent SafeRL that can achieve good performance while respecting constraints, though each has trade-offs in terms of safety guarantees vs. efficiency.
- SafeMARL is a frontier with significant potential impact (e.g., fleet management, multi-robot systems). Early algorithms like MACPO and shielding strategies show feasibility, but general solutions for decentralized and competitive scenarios are still lacking.

We identified several open research problems that require further work: from guaranteeing zero violations to handling partial observability, and from fully decentralized safe coordination to safe learning in non-stationary multi-agent environments. These problems underscore that SafeRL is not a solved problem—there are theoretical challenges (ensuring safety and convergence), practical issues (scalability, function approximation errors), and new frontiers (multi-agent interactions). In conclusion, SafeRL offers a pathway to more trustworthy AI systems by marrying reinforcement learning with constraint satisfaction. As RL agents become more capable and autonomous, ensuring their safety will be paramount. We hope this survey serves as a useful resource for researchers to understand the current landscape and to inspire further advances. The continued development of SafeRL methods will help unlock applications of RL in domains that are currently out of reach due to safety concerns, ultimately enabling AI to make beneficial decisions without posing undue risk.

References

- J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 22–31. PMLR, 2017.
- Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024. URL <https://www.marl-book.com>.
- M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018a.
- Moayad Alshiekh, Roderick Bloem, Richard Ehlers, Bernhard Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018b.
- E. Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.
- D. Amodei, C. Olah, J. Steinhardt, and etal. Concrete problems in AI safety. *arXiv:1606.06565*, 2016.
- Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv*, 2017.
- F. Berkenkamp and A. Schoellig. Safe and robust learning control with gaussian processes. In *Proceedings of the European Control Conference (ECC)*, 2015.
- F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems 30*, 2017.

- Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4917–4924, 2016. doi: 10.1109/ICRA.2016.7487720.
- D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.
- E. Bitar and S.H. Low. Deadline differentiated pricing of deferrable electric loads. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 4064–4069, 2012. doi: 10.1109/CDC.2012.6426466.
- L. Blackmore, M. Ono, B. Williams, and R. Siegwart. A probabilistic approach to optimal robust path planning with obstacles. In *American Control Conference (ACC)*, pages 7–12, 2010.
- V. S. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2):294–311, 2002.
- V. S. Borkar and R. Jain. Risk-constrained markov decision processes. *IEEE Transactions on Automatic Control*, 59(9):2574–2579, 2014.
- Baiming Chen, Yu Zheng, Yaodong Yang, Yisong Yue, and Jun Wang. Context-aware safe reinforcement learning for non-stationary environments. In *arXiv preprint arXiv:2101.00531*, 2021.
- X. Chen, D. Simchi-Levi, and Y. Wei. Risk-averse stochastic inventory control: A perspective on approximate dynamic programming. *Operations Research*, 62(6):1302–1317, 2014.
- Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-sensitive and robust decision-making: A cvar optimization approach. In *Advances in Neural Information Processing Systems*, volume 28, 2015.
- Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Lyapunov-based safe policy optimization for continuous control. In *Advances in Neural Information Processing Systems 31*, 2018a.
- Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems 30*, 2018b.
- Hyeonuk Chung, Taejin Kim, Donghwan Kim, and Kyoungchul Kim. Multi-objective reinforcement learning for sustainable manufacturing process control. *IEEE Transactions on Automation Science and Engineering*, 17(4):2073–2084, 2020. doi: 10.1109/TASE.2020.2985871.
- S. Curi, A. Krause, and F. Berkenkamp. Learning safe policies with expert guidance. In *Proc. of the 37th International Conference on Machine Learning*, 2020.
- G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa. Safe exploration in continuous action spaces. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.
- Emiliano Dall’Anese, Gil Zussman, and Georgios B. Giannakis. Chance-constrained ac optimal power flow for distribution system operations. *IEEE Transactions on Smart Grid*, 6(6):2890–2901, 2015. doi: 10.1109/TSG.2015.2415668.
- I. ElSayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng. Safe multi-agent reinforcement learning via shielding. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 483–491, 2021.
- Tom Everitt, Victoria Krakovna, and Shane Legg. Safe strategies for agent modelling in games. *arXiv preprint arXiv:1901.03258*, 2019.
- J. Filar, D. Krass, and K. Ross. Risk-sensitive control of markov decision processes. *Mathematics of Operations Research*, 20(1):128–162, 1995.

- Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.
- Sam Ganzfried. Safe equilibrium. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 5230–5236, 2022.
- J. Garcia and F. Fernandez. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.
- P. Geibel and F. Wysotski. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.
- O. Gottesman, F. Johansson, J. Meier, S. Ding, I. Li, A. Faisal, and D. Sontag. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, 2019.
- Djordje Grbic and Sebastian Risi. Safe reinforcement learning through meta-learned instincts. In *Proceedings of the Artificial Life Conference (ALIFE)*, 2020. URL https://direct.mit.edu/isal/article/doi/10.1162/isal_a_00318/98468.
- S. Gu, J. G. Kuba, Y. Chen, Y. Du, L. Yang, A. Knoll, and Y. Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319:103905, 2023.
- S. Gu, B. Sel, Y. Ding, L. Wang, Q. Lin, A. Knoll, and M. Jin. A review of safe reinforcement learning: Methods, theory and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- M. Heger. Consideration of risk in reinforcement learning. In *Proc. of the 11th International Conference on Machine Learning*, 1994.
- H. Huang, Y. Chow, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. In *arXiv preprint arXiv:1805.01677*, 2018.
- N. Jiang, J. K. Lee, A. Thomas, F. Yue, and D. Sontag. Improving clinical interpretability and safety of reinforcement learning. In *Proceedings of the Machine Learning for Healthcare Conference*, 2019.
- R. Jiang, J. Wang, and Y. Guan. Risk-constrained optimal power flow with probabilistic guarantees. In *IEEE Power and Energy Society General Meeting*, 2012.
- Moutaz Khouja. The economic production lot size model with safety constraints. *Omega*, 31(1): 47–54, 2003. doi: 10.1016/S0305-0483(02)00067-X.
- B. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. Yogamani, P. Perez, P. Frossard, and D. K. An. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.
- M. Komorowski, A. Celi, O. Badawi, L. Gordon, and A. Faisal. The artificial intelligence clinician: Improving intensive care unit treatment with reinforcement learning. *Nature Medicine*, 24:1716–1720, 2018.
- Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Constrained policy optimization with explicit behavior density for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 13329–13341, 2021.
- Xueguang Lyu and Lantao Liu. Constrained policy optimization via penalized td3. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4362–4368, 2021.
- A. Majumdar and M. Pavone. Should robots be risk-averse? risk-averse trajectory optimization via cvar. *International Journal of Robotics Research*, 39(4):429–446, 2020.

- P. A. Mattei and P. Gopalan. Leveraging risk-sensitive and distributional reinforcement learning for sepsis treatment. In *Machine Learning for Healthcare Conference (MLHC)*, 2019.
- M. Ono and B. Williams. Chance-constrained markov decision processes with application to risk-limiting operation of spacecraft. *Autonomous Robots*, 35:365–377, 2013.
- M. Ono and B. Williams. Chance-constrained dynamic programming with application to risk-aware robotic spacecraft guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.
- OpenAI. Safety starter agents, 2019. URL <https://github.com/openai/safety-starter-agents>.
- L. A. Prashanth. Policy gradients for cvar-constrained mdps. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 155–169, 2014.
- L. A. Prashanth and M. Ghavamzadeh. Variance-constrained actor-critic algorithms for risk-sensitive reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 353–361, 2016.
- Anirudh Raghu, Matthieu Komorowski, Leo Anthony Celi Singh, Peter Szolovits, and Finale Doshi-Velez. Model-based reinforcement learning for sepsis treatment. In *Machine Learning for Health (ML4H) Workshop at NeurIPS 2017*, 2017.
- H. Ravanbakhsh, S. Bastani, and S. K. S. Gupta. Learning risk-aware scheduling policies for cloud datacenters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.
- A. Ray, J. Achiam, and D. Amodei. Benchmarking Safe Exploration in Deep Reinforcement Learning. *arXiv:1910.01708*, 2019.
- R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.
- Dorsa Sadigh, Shankar Sastry, S. Shankar Seshia, and Anca Dragan. Planning for autonomous cars that leverage effects on human actions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2016.
- W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans. Trial without error: Towards safe reinforcement learning via human intervention. In *Proc. of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2018.
- William Saunders, Girish Sastry, Andreas Stuhlmüller, and Owain Evans. Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.
- Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1710.03473*, 2017.
- A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2nd edition, 2014.
- Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge university press, 2008.
- S. Singh, A. Majumdar, and M. Pavone. Risk-sensitive trajectory optimization for autonomous driving. In *Robotics: Science and Systems (RSS)*, 2018.
- A. Tamar, Y. Glassner, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems 28*, 2015.

- C. Tessler, D. Givoli, and S. Mannor. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2019.
- B. Thananjeyan and et al. Safety augmented value estimation for legged robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2021.
- M. Turchetta, F. Berkenkamp, and A. Krause. Safe exploration in finite MDPs with gaussian processes. In *Advances in Neural Information Processing Systems 29*, 2016.
- K. P. Wabersich and M. N. Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135, 2018.
- A. Wachi, X. Shen, and Y. Sui. A survey of constraint formulations in safe reinforcement learning. In *Proc. of the 33rd International Joint Conference on Artificial Intelligence (Survey Track)*, 2024.
- Gerhard Weiss. *Distributed Artificial Intelligence: A Modern Approach*. MIT press, 1996.
- Gerhard Weiss. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT press, 1999.
- Michael Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2009.
- Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J. Ramadge. Projection-based constrained policy optimization. In *International Conference on Learning Representations (ICLR)*, 2020a. URL <https://arxiv.org/abs/2010.03152>.
- Yaodong Yang, Junjie Luo, Minne Li, Ying Wen, Weinan Wei, Zhen Liu, and Jun Wang. Multi-agent reinforcement learning is a game with strategy-proof information design. In *Advances in Neural Information Processing Systems*, volume 33, pages 18392–18403, 2020b.
- Z. Yang, Y. Xie, B. Zhang, and C. Zhang. Towards safe reinforcement learning for multi-agent systems in multi-stage games. In *Advances in Neural Information Processing Systems 34*, 2021.
- L. Zhang, L. Li, W. Wei, H. Song, Y. Yang, and J. Liang. Scalable constrained policy optimization for safe multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.
- P. Zhang, K. Wang, and L. Liu. Probabilistic load flow calculation using cumulant method considering correlation of wind power output. *IEEE Transactions on Power Systems*, 26(4): 2543–2551, 2011.
- Z. Zhang, R. Fox, and R. Tedrake. Risk-averse motion planning under uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- Z. Zheng and S. Gu. Safe multi-agent reinforcement learning with bilevel optimization in autonomous driving. *IEEE Transactions on Artificial Intelligence*, 2024.
- Weichao Zhou and Wenchao Li. Safety-aware apprenticeship learning, 2018. URL <https://arxiv.org/abs/1710.07983>.
- Y. Zhu, M. Chen, and L. Sun. Sim2real transfer for autonomous driving with control variability. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.