Feasible Reachable Policy Iteration

Shentao Qin [*1] **Yujie Yang** [*1] **Yao Mu** [*2] **Jie Li** [1] **Wenjun Zou** [1] **Jingliang Duan** [3] **Shengbo Eben Li** [1]

Abstract

The goal-reaching tasks with safety constraints are common control problems in real world, such as intelligent driving and robot manipulation. The difficulty of this kind of problem comes from the exploration termination caused by safety constraints and the sparse rewards caused by goals. The existing safe RL avoids unsafe exploration by restricting the search space to a feasible region, the essence of which is the pruning of the search space. However, there are still many ineffective explorations in the feasible region because of the ignorance of the goals. Our approach considers both safety and goals; the policy space pruning is achieved by a function called feasible reachable function, which describes whether there is a policy to make the agent safely reach the goals in the finite time domain. This function naturally satisfies the self-consistent condition and the risky Bellman equation, which can be solved by the fixed point iteration method. On this basis, we propose feasible reachable policy iteration (FRPI), which is divided into three steps: policy evaluation, region expansion, and policy improvement. In the region expansion step, by using the information of agent to reach the goals, the convergence of the feasible region is accelerated, and simultaneously a smaller feasible reachable region is identified. The experimental results verify the effectiveness of the proposed FR function in both improving the convergence speed of better or comparable performance without sacrificing safety and identifying a smaller policy space with higher sample efficiency.

[*]Equal contribution [1]School of Vehicle and Mobility, Tsinghua University, Beijing, China [2]Department of Computer Science, The University of Hong Kong, Hong Kong, China [3]School of Mechanical Engineering, University of Science and Technology Beijing, Beijing, China. Correspondence to: Shengbo Eben Li <lishbo@tsinghua.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

The goal-reaching tasks with safety constraints are a very common class of control problem in real-world applications of Reinforcement Learning (RL). (Tessler et al., 2018; Andrychowicz et al., 2020; Altman, 2021; Duan et al., 2024). In many safe exploration problems, the existence of safety leads to the exploration being terminated when the constraint is violated, and the existence of goals leads to the sparsity of rewards, both of which lead to the invalidity of many exploratory samples. We urgently need an efficient sample solution to solve the exploration difficulties of constrained goal-reaching problems.

Safe reinforcement learning (Safe RL) is designed to solve the optimal control problem (OCP) with safety constraint (Achiam et al., 2017; Tessler et al., 2019; Yang et al., 2023b). The safe RL expects to find an optimal feasible policy, where feasibility means the agent will never violate constraint during the trajectories.

The mainstream safe RL algorithms constrains the policy optimization process to the feasible region, which solves the oscillation problem of the Lagrangian method (Liu & Tomizuka, 2014; Achiam et al., 2017; Li, 2023; Yu et al., 2022). The essence of a feasible region is pruning the policy search space. However, it takes quite a long time to identify the feasible region through fixed point iteration, which involves feasibility evaluation of the infinite time domain, resulting in slow convergence. Besides, although all states in feasible regions meet the safety constraints, a considerable part of them cannot reach the goal in the finite time domain, which causes inefficiency of exploration. This part of the region can be further pruned to speed up the training convergence. How to further prune the feasible region and improve the sample efficiency of safe RL algorithms is still an open problem.

From the analysis above, only a tiny subset of state space is valuable to explore, where the target state can be reached in a finite horizon, and the safety constraints can be persistently satisfied. Based on this idea, the trajectories generated by policy in state space can be divided into four categories: 1) trajectories that successfully reach the target set without violating constraints, 2) trajectories that can reach the target set but violate constraints, 3) trajectories that can never reach the target set but never violate the constraints, and 4)

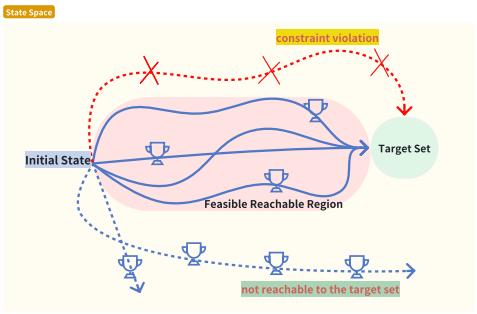


Figure 1. The four categories of trajectories in state space. We focus on the trajectories belonging to feasible reachable region.

trajectories that neither reach the target nor guarantee persistent constraint satisfaction. We denote the first category as feasible reachable, the second as infeasible, the third as unreachable, and the fourth as both infeasible and unreachable. As shown in Fig. 1 , only the trajectories belonging to the feasible reachable region (FR region) are needed. To this end, we propose a feasible reachable function that identifies the feasible reachable set of target space and restricts environmental exploration and policy improvement to this set. Our main contributions are:

- We propose a novel feasible reachable function (FR function), which describes whether there is a policy to safely reach the target set. Our method takes both feasibility related to safety constraints and reachability related to goals into account, identifying the FR region to limit exploration. Our function naturally satisfies the self-consistent condition and the risky Bellman equation, which enables it to be solved by the fixed point iteration method.
- We propose a safe RL algorithm called feasible reachable policy iteration (FRPI), which uses the FR function to restrict policy improvement in the FR region to avoid inefficient exploration that is neither feasible nor reachable. The algorithm is divided into three steps: policy evaluation, region expansion, and policy improvement. In the region expansion step, the convergence of the feasible region is accelerated by using goal-reachability information, and simultaneously a smaller feasible reachable region is identified.
- We test our algorithm on the frozen lake (gym) environment, two classical control tasks, and the safety gym benchmark. The experimental results verify that our algorithm achieves higher sample efficiency than baselines while maintaining better or comparable performance without sacrificing safety. Further analysis shows that the FR function can effectively accelerate policy space pruning, and identify a smaller FR region compared with existing methods.

2. Related Work

Safe RL problems have gained growing attention due to the safety requirements in the practical applications of RL. Safe RL is usually formulated as a constrained Markov decision process (Brunke et al., 2022; Ma et al., 2022). Generally, we divide the safe RL approaches into two categories (Li, 2023), direct and indirect methods. In the first category, the constrained OCP is viewed as a constrained optimization problem, and its optimum must be found by proper constrained optimization algorithms. In the second category, the constrained Bellman equation should be built first, which is the sufficient and necessary optimality condition, and its solution is then calculated as the optimal policy.

Direct Methods solve constrained OCPs using constrained optimization algorithms: penalty function methods (Guan et al., 2022), Lagrangian methods (Chow et al., 2018a), trust-region methods (Achiam et al., 2017; Schulman et al., 2015), other approaches such as conservative updates (Bharadhwaj et al., 2020) and so on. These algorithms expect discounted accumulative costs below a hand-crafted threshold but suffer from unstable training processes and constant constraint violations. For example, the Lagrangian methods have violent oscillation because the Lagrange multiplier does not provide any guarantee on rewards or costs of intermediate policies (Peng et al., 2022).

Indirect Methods usually explicitly learn a feasible region (Liu & Tomizuka, 2014; Ames et al., 2019) to identify the feasibility of policy, which solves the oscillation problem of the direct method. Feasible region are usually represented by safety certificates, for example, energy functions such as control barrier function (CBF) (Luo & Ma, 2021; Ames et al., 2019; Yang et al., 2023b), Lyapunov functions (Chow et al., 2018b; Richards et al., 2018; Chang et al., 2019), safety index (SI) (Liu & Tomizuka, 2014; Ma et al., 2022), other certificates like Hamilton-Jacobi reachability function (Chen et al., 2021; Yu et al., 2022; Zheng et al., 2024) and constraint decay function (CDF) (Yang et al., 2023d;c). (Yang et al., 2021) introduces the WCSAC and optimizes policies under the premise that their worst-case performance satisfies the constraints. (Yang et al., 2023a) introduces the distributional safety critic module and points out that sample efficiency is particularly crucial in safety-critical problems, and off-policy RL is a natural approach to solving safe RL problems.

The core idea of energy function is that the safe energy of a dynamical system dissipates when it is approaching the safe region. The main limitation of exiting energy functions is that they are too conservative to find the maximum feasible region. The recent indirect methods try to learn one policy tackling safety and optimality simultaneously. (Hsu et al., 2021) introduces the EFPPO to solve the stabilize-avoid problem. (Yu et al., 2022) introduces the RAC, which re-

alizes rigorous zero-violations of cost. (So & Fan, 2023) firstly introduces the reach-avoid Q-learning algorithm to solve the RA problem. (Yang et al., 2023d) introduces CDF and find the largest feasible region. Different from the above work, we focus on getting both the largest feasible region and best sample efficiency simultaneously by proper pruning of policy space, which makes the policy iteration more efficient.

3. Preliminary

3.1. Problem formulation

We consider a deterministic Markov decision process (MDP) specified by a tuple $(\mathcal{X}, \mathcal{U}, f, r, \gamma, d_{\text{init}})$, where $\mathcal{X} \subseteq \mathbb{R}^n$ is the state space, $\mathcal{U} \subseteq \mathbb{R}^m$ is the action space, $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ is the dynamics model, $r : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$ is the reward function, $0 < \gamma < 1$ is the discount factor, and d_{init} is the initial state distribution. The goal-reaching problems have a target set, which is a subset of state space, denoted as X_{goal}.

Based on the target set, we can give the math formulation of goal-reaching problem as followed:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{x_0 \sim d_{\text{init}}(x)} \left\{ \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t) \right\}, \\ \text{s.t.} \quad & h(x_t) \leq 0, t = 0, 1, \ldots, T, \\ & g(x_T) = 1, \\ & T < C, \end{aligned} \tag{1}$$

where $C \in \mathbb{N}^+$, $h : \mathcal{X} \to \mathbb{R}$, is the constraint function. Our aim is to find the optimal feasible reachable policy $\pi : \mathcal{X} \to \mathcal{U}$, which maximizes the expected cumulative rewards. Reachability is specified through goal identification function $g : X \to \{0, 1\}$,

$$g(x) = \begin{cases} 1 & x \in \mathrm{X}_{goal}, \\ 0 & x \notin \mathrm{X}_{goal}. \end{cases}$$

In safe RL, we focus on the *persistent* safety (Li, 2023; Yu et al., 2022; Yang et al., 2023d) instead of the *temporary* safety . In this regard, the feasible set is defined as the set of states which can be safe persistently. The detail is seen in Appendix A.1. However, only exploring the feasible region may cause the agent never to reach the target set forever, so we combine the feasibility and reachability to identify the feasible reachable region where the sample is efficient for agent.

4. Feasible Reachable Region

In this section, we discuss the relationship between feasible regions and reachable regions, as shown in Fig. 2.

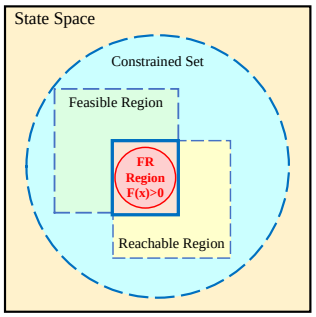


Figure 2. The intuitive relationship among the state space. Constrained set, feasible region, reachable region, feasible reachable region (FR Region). $\mathrm{X}^*_{\text{FR}} \subseteq (X^*_{\text{feas}} \cap \mathrm{X}^*_{\text{reach}})$.

Definition 4.1 (Reachable Region)**.**

1) A state x is reachable if there exists a policy π and a time $T < C$, such that $g(x_T) = 1$, where the successive state is sampled by the policy π.

2) A policy is reachable in state x if there exists a time $T < C$ such that $g(x_T) = 1$.

3) The reachable region of π, denoted as $\mathrm{X}^{\pi}_{\text{reach}}$, is the set of all states in which π is reachable. The unreachable region of π is $(\mathrm{X}^{\pi}_{\text{reach}})^c = \mathcal{X} \setminus X^{\pi}_{\text{reach}}$.

4) The maximum reachable region, denoted as $\mathrm{X}^*_{\text{reach}}$, is the set of all reachable states. The unreachable region is $(\mathrm{X}^*_{\text{reach}})^c = \mathcal{X} \setminus \mathrm{X}^*_{\text{reach}}$.

Based on this, the definition of feasible reachable region and feasible reachability identification function are given as follows.

Definition 4.2 (Feasible Reachable Region)**.**

1) A state is feasible reachable if there exists a policy π and a time $T < C$, such that $g(x_T) = 1$, and $h(x_t) \leq 0$, $t = 0, 1, \ldots, T$

2) The feasible reachable region, denoted as $\mathrm{X}^{\pi}_{\text{FR}}$, is the set of all states in π is feasible reachable. The feasible reachable region under policy is: $\mathrm{X}^{\pi}_{\text{FR}} \triangleq (X^{\pi}_{\text{feas}} \cap \mathrm{X}^{\pi}_{\text{reach}})$.

3) The maximum feasible reachable region, denoted as X^*_{FR}, is the set of all feasible reachable states, i.e., $\mathrm{X}^*_{\text{FR}} \triangleq \cup_{\pi} X^{\pi}_{\text{FR}}$.
We have $\mathrm{X}^*_{\text{FR}} \subseteq (\mathrm{X}^*_{\text{feas}} \cap \mathrm{X}^*_{\text{reach}})$ hold. (The proof can be found in Appendix A.2)
Notice: In this paper, we denote X^*_{FR} as X^*, and denote $\mathrm{X}^{\pi}_{\text{FR}}$ as X^{π} for simplicity.
We require that any initial state sampled from d_{init} is feasible reachable so that problem (1) has a solution. The maximum feasible reachable region is the largest area where the policy can reach the target set without constraint violations in an infinite horizon.

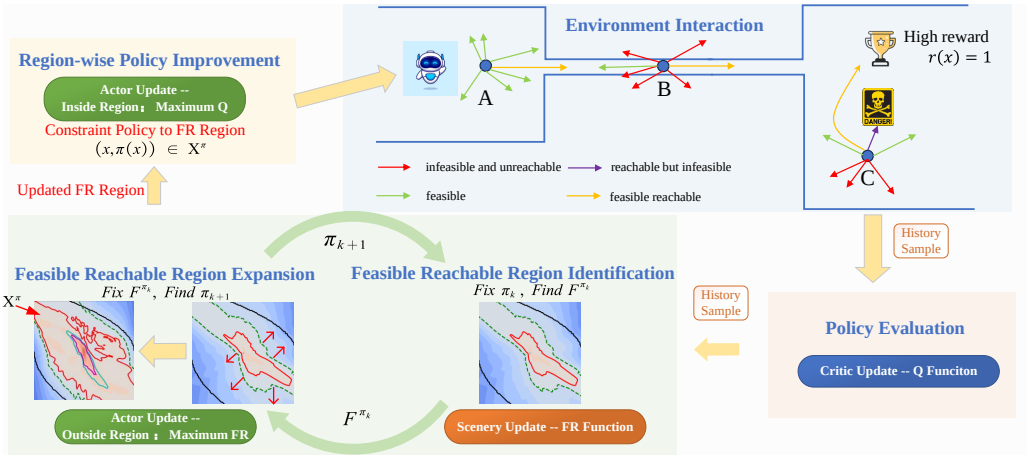


Figure 3. The FRPI Algorithm is divided into three steps: policy evaluation, region expansion, and policy improvement. We update scenery and critic simultaneously in policy evaluation and make the FR region expansion. Finally, we make the region-wise policy improvement in the FR region.

Directly solving feasibility and reachability problems is still challenging because it has a great many states in trajectories to identify. However, the feasible reachable function aggregates the awkwardly many state evaluations into a single one, making the problem tractable. We define the feasible reachable function as follows.

Definition 4.3 (Feasible Reachability Identification)**.** A function $F^{\pi}\colon \mathcal{X} \to \mathbb{R}$ is a feasible reachability identification function of a policy π if $\forall x \in \mathcal{X}$,

$$F^{\pi}(x) > 0 \iff h(x_t) \leq 0, g(x_T) = 1,$$

where $x_0 = x$ and $\{x_t\}_{t=1}^{T}$ are sampled by π.

The specific formulation of this concept will be detailed in the next section. In particular, the concepts of FR action and FR policy are important for understanding feasible reachable, which is defined as follows.

Definition 4.4 (FR Action and FR Policy)**.**

1) Feasible reachable action set under policy
$\mathrm{U}^{\pi} = \{u \mid f(x,u) \in \mathrm{X}^{\pi}, x \in \mathrm{X}^{\pi}\}$

2) Maximum feasible reachable action set
$\mathrm{U}^{*} \triangleq \{u \mid f(x,u) \in \mathrm{X}^{*}, x \in \mathrm{X}^{*}\}$

3) Feasible reachable policy set is defined as follows:
$\Pi^{*} \triangleq \{\pi \mid u = \pi(x), u \in \mathrm{U}^{*}, x \in \mathrm{X}^{*}\}$

5. Method and Theoretical Analysis

In this section, we propose our feasible reachable policy iteration (FRPI), a highly sample efficient algorithm. First, we introduce a feasible reachable function (FR function), which naturally satisfies the self-consistent condition and the risky Bellman equation. Then, we detail the region-wise policy improvement and prove the region expansion. Finally, we present the FRPI algorithm and prove the convergence of algorithm. As shown in Fig. 3, our algorithm contains three modules to train: Actor, Critic, and Scenery. In FRPI, we train these modules in three phases: policy evaluation, region identification, and region-wise policy improvement (region expansion and policy improvement). At the start of training, we first collect data containing four sorts of trajectories, in which only the feasible reachable trajectory is desired. Our algorithm is built in an off-policy context to obtain higher sample efficiency. From the perspective of algorithm update, it can be divided into the following steps:

1. **Policy Evaluation**. We update the critic module, the Q-function, by the Q self-consistency condition, which can give the expected return of the current policy.

2. **FR Region Identification**. We update the scenery module, the FR function, by introducing the FR self-consistency condition, which can identify the feasible reachable region of the current policy.

3. **FR Region Expansion and Policy Improvement**. We update the actor module by region-wise policy improvement. For the state outside the FR region, we maximize the scenery function to get a more feasible policy with a greater feasible reachable region; for the state inside the region, we maximize the value function to get a higher return policy.

Convergence of FRPI requires convergence of both regions and policies. The core step is the learning process of region, which contains both region identification and region expansion. In the FR region identification, we fix the policy function and update the FR function by feasible reachable self-consistent condition. In the FR region expansion, we fixed the scenery function to find a better policy by promoting feasible reachability outside the region. Once the FR region is given, we can efficiently improve the return

within the region to find the optimal policy that satisfies the feasible reachability constraint.

5.1. Feasible Reachable Function

Definition 5.1 (FR Function)**.** For any policy, we can define its FR function, which is specifically crafted to determine its feasible reachable region.

$$F^{\pi}(x) = \begin{cases} \gamma^{N_g} & N_g < N_c, \\ -\gamma^{N_c} & N_c < N_g, \\ 0 & N_c = \infty, N_g = \infty, \end{cases} \tag{2}$$

where N_g is the step to reach the goal, N_c is the step to violate the constraints. The expansion of the above equation is as follows:

$$F^{\pi}(x_0) = g(x_0) + c(x_0) + \sum_{m=1}^{T} \prod_{n=0}^{m-1} (1 + c(x_n))(1 - g(x_n))\gamma^n (g(x_m) + c(x_m)),$$

where we define $g(x) = \mathbf{1}_{x_{\text{goal}}}(x)$, indicating whether the target set is reached, $c(x) = -\mathbf{1}_{\bar{x}_{\text{cstr}}}(x)$, indicating whether a state constraint is violated.

Only if the policy reaches the goal without violating any constraints from the initial state in a finite number of steps will the value of $F^{\pi}(x)$ be positive, thereby satisfying the definition of feasible reachable. Besides, N_g represents the distance from the current state to the target state. If the policy reaches the goal in less time, the value of $F^{\pi}(x)$ will be higher. To justify the choice of FR function to represent the feasible reachable region, we first prove that it is a feasibility reachable function.

Proposition 5.1 The FR function is a feasible reachability identification function. Additionally, the zero-superlevel set is feasible reachable region, i.e., $\{x \in \mathcal{X} | F^{\pi}(x) > 0\} = \mathrm{X}^{\pi}$, and the zero-level set is feasible region, i.e., $\{x \in \mathcal{X} | F^{\pi}(x) = 0\} = \mathrm{X}^{\pi}_{feas}$.

Proof. See Appendix B.1.

The following proposition tells us that the optimal FR function represents the maximum feasible region.

Definition 5.2 (Optimal FR Function)**.** The optimal feasible reachable function $F^* : \mathcal{X} \to \mathbb{R}$ is defined as

$$F^*(x) = \max_{\pi} F^{\pi}(x). \tag{3}$$

Proposition 5.2. The zero-superlevel set of the optimal FR function is the maximum feasible reachable region, i.e., $\{x \in \mathcal{X} \mid F^*(x) > 0\} = \mathrm{X}^*$.

Proof. See Appendix B.2.

As shown in Fig. 4, we can solve the goal-reaching problem with safety constraints by feasible reachable policy tree identified by the FR function.

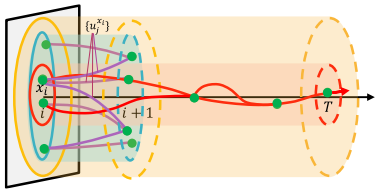


Figure 4. Forward Feasible Reachable Policy Tree by FR Function Identification. The green zone represents the feasible policy, while the red zone represents the FR policy. For a given state, actions are constrained to the feasible reachable action set.

Theorem 5.3 (Self-Consistency Condition of FR Function)**.** *The FR function satisfies the self-consistency condition for all x in $\mathcal{X}$*

$$F^{\pi}(x) = g(x) + c(x) + (1 - g(x))(1 + c(x))\gamma F^{\pi}(x'). \tag{4}$$

Proof. See Appendix B.3.

The right-hand side of (4) can be viewed as an operator mapping from a function to another, and F^{π} is a fixed point of this mapping. The following theorem shows that this mapping is a contraction mapping on a complete metric space, therefore there is a unique fixed point.

Theorem 5.4 (The Unique Fixed Point)**.**

Define the feasible reachable identification operator D^{π} as

$$D^{\pi}(F(x)) = g(x) + c(x) + (1 - g(x))(1 + c(x))\gamma F(x').$$

We can show the operator D^{π} has a unique fixed point F^{π}.

Proof. See Appendix B.4.

The fixed point F^{π} can be found by iteratively applying D^{π} starting from an arbitrary F, which is shown in feasible reachable region identification algorithm as follows. According to Banach's fixed-point theorem, F_k converges to F^{π}, i.e., $\lim_{k \to \infty} F_k = F^{\pi}$.

Algorithm 1 Feasible Reachable Region Identification

Input: initial FR function F_0, policy π.
for *each iteration* k **do**
  for *each state* $x \in X$ **do**
    $F_{k+1}(x) = g(x) + c(x) + (1 - g(x))((1 + c(x))\gamma F_k(x'))$
  end
end

5.2. Region-wise Policy Improvement

Our proposed algorithm, feasible reachable policy iteration (FRPI), involves alternating three steps: policy evaluation,

feasible reachable region identification, and region-wise policy improvement. Region-wise policy improvement is the core step of FRPI, where we expand the feasible reachable region and increase the state-value function to the greatest extent in each policy update.

Inside the feasible reachable region of π_k, we solve a constrained optimization problem, i.e., $\forall x \in \mathrm{X}^{\pi_k}$,

$$\begin{aligned} &\pi_{k+1}(x) = \arg\max_u r(x,u) + \gamma V^{\pi_k}(x'), \\ &\text{s.t. } F^{\pi_k}(x') > 0. \end{aligned} \tag{5}$$

The constraint in (5) requires that the next state is still in X^{π_k}.
Outside the feasible reachable region of π_k, we maximize the FR function of the next step without constraints, i.e., $\forall x \in X^{\pi_k}$,

$$\pi_{k+1}(x) = \arg\max_u F^{\pi_k}(x'). \tag{6}$$

Next, we prove that the above update rule results in a larger feasible reachable region and a greater state-value function.

Theorem 5.5 (FR Region Expansion)**.** *In a deterministic MDP, the FR region of π_{k+1}, denoted as $\mathrm{X}^{\pi_{k+1}}$, is greater than or equal to the FR region of π_k, denoted as X^{π_k}.*

Proof. See Appendix B.5.

From the proof of Theorem 5.5, we can see that both (5) and (6) play important roles in ensuring the monotonicity of the FR region expansion. Equation (6) takes a maximization so that the FR function increases outside the feasible region. Equation (5) constraints the FR region at the value of zero-superlevel set of FR function.

Theorem 5.6 (FR Region Expansion)**.** *In a deterministic MDP, the FR region of π_{k+1}, denoted as $\mathrm{X}^{\pi_{k+1}}$, is greater than or equal to the FR region of π_k, denoted as X^{π_k}.*

Proof. See Appendix B.6.

5.3. Feasible Reachable Policy Iteration

In this subsection, we prove FRPI converges to a policy with the optimal FR function and the optimal state-value function, which represents the maximum feasible reachable region. First of all, we consider the optimal state-value function in feasible reachable region, which is defined as follows.

Definition 5.7 (Optimal State-Value Function)**.** The optimal state-value function $V^*\colon \mathrm{X}^* \to \mathbb{R}$ is defined as

$$V^*(x) = \max_{\pi \in \Pi^*} V^{\pi}(x). \tag{7}$$

The optimal state-value function satisfies a recursive relationship called the feasible reachable Bellman equation, which is also a necessary and sufficient condition of the optimal state-value function.

Theorem 5.8 (Feasible Reachable Bellman Equation)**.** *The state-value function $V : \mathrm{X}^* \to \mathbb{R}$ is optimal if and only if it satisfies the feasible reachable Bellman equation for all x in X^**

$$V(x) = \max_{u \in \mathrm{U}^*(x)} r(x,u) + \gamma V(x'). \tag{8}$$

The feasible reachable Bellman equation also represents the convergence of V, which means $\lim_{k\to\infty} V^{\pi_k} = V^*$, where V^* is the optimal state-value function. Secondly, we introduce a recursive relationship of the optimal FR function called the risky Bellman equation, which is also a necessary and sufficient condition of the optimal FR function.

Proof. See Appendix B.7.

Secondly, we introduce a recursive relationship of the optimal FR function called the risky Bellman equation, which is also a necessary and sufficient condition of the optimal FR function.

Theorem 5.9 (Risky Bellman Equation)**.** *The FR function $F\colon \mathcal{X} \to \mathbb{R}$ is the optimal FR function if and only if it satisfies the risky Bellman equation for all x in $\mathcal{X}$*

$$F(x) = c(x) + g(x) + (1 + c(x))(1 - g(x))\gamma \max_u F(x'). \tag{9}$$

Combined with Definition 5.5, the risky Bellman equation also represents the convergence of F in region expansion, i.e., $\lim_{k\to\infty} F^{\pi_k} = F^*$, where F^* represent the maximum feasible reachable region.

Proof. See Appendix B.8.

Algorithm 2 Feasible Reachable Policy Iteration (FRPI)

Input: initial policy π_0.
for *each iteration k* **do**
  Compute F^{π_k} using FR region identification;
  Compute V^{π_k} using policy evaluation;
  for *each state $x \notin X^{\pi_k}$* **do**
    $\pi_{k+1}(x) \leftarrow \arg\max_u F^{\pi_k}(x')$
  end
  for *each state $x \in X^{\pi_k}$* **do**
    $\pi_{k+1}(x) \leftarrow \arg\max_u \{r(x,u) + \gamma V^{\pi_k}(x')\}$
    subject to $F^{\pi_k}(x') > 0$
  end
end

Finally, we prove the convergence of FRPI by proving that the FR function and the state-value function converge to the solutions of their corresponding Bellman equations.

Theorem 5.10 (Convergence of FRPI)**.** *Suppose that at the k-th iteration, for all x in $\mathcal{X}$, $F^{\pi_{k+1}}(x) = F^{\pi_k}(x)$, and for all x in X^*, $V^{\pi_{k+1}}(x) = V^{\pi_k}(x)$. Then, it follows that $F^{\pi_k} = F^*$ and $V^{\pi_k} = V^*$. Moreover, convergence can be achieved within a finite number of iterations in finite state and action spaces.*

Proof. See Appendix B.9.

6. Experiments

We seek to answer the following questions through our experiments:
1. Can FR function enable an efficient policy space pruning to achieve faster convergence than other algorithms?
2. Can FRPI-SAC speed up feasible region expansion, and simultaneously identify a smaller FR region?
3. Do FRPI-SAC achieve a comparable performance faster than other algorithms without sacrificing safety?

6.1. Practical Implementation

We give the implementation of FR function and integrate it with SAC, the details is seen in Appendix C.

6.2. Baselines

For comparison, we adopt SAC-Lag(Ha et al., 2021) or penalty function methods as baseline of direct method. In particular, we adopt the FPI (Yang et al., 2023d) as the latest baseline of indirect method.

We also adopt the SAC (Haarnoja et al., 2018) as a constraint-free baseline for providing a performance upper bound, to verify the comparable performance of FRPI.

6.3. Policy Space Prunning

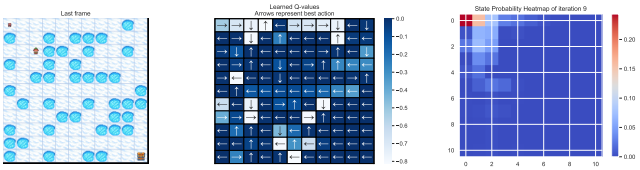
Frozen lake involves crossing an iced lake from start to goal without falling into any holes. The player may not always move in the intended direction due to the slippery nature of the frozen lake. The reward in the frozen lake environment is very sparse. We compared penalty function method, FPI, and FRPI in this environment. As shown in Fig. 5, the results show that our pruning of the policy space is the most efficient, and an accurate Q is learned through the FR region identified by the FR function.

6.4. FR Region Expansion

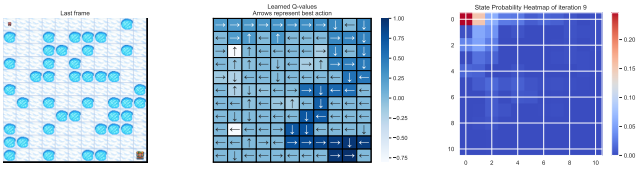
With regard to the question (2), we test our proposed method on two classical control tasks where the dynamics are simple and known accurately, and thus we can get the visible maximum FR regions.

(1) Adaptive Cruise Control (ACC) The goal of ACC is to control a following vehicle to converge to a fixed distance with respect to a leading vehicle, as illustrated in Fig. 6(a). The settings of the system is seen in Appendix D.1.

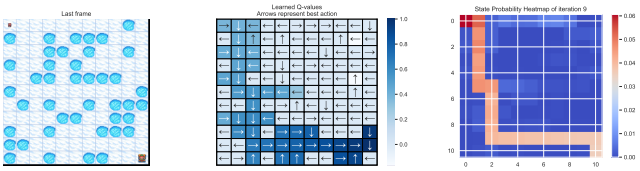
(2) Quadrotor Trajectory Tracking Different from the previous stabilization task, the Quadrotor is a trajectory tracking task that comes from safe-control-gym (Yuan et al., 2022), where a 2D quadrotor is required to follow a circular trajectory in the vertical plane while keeping the vertical



(a) The distribution of states and actions by Q-learning(Penalty)

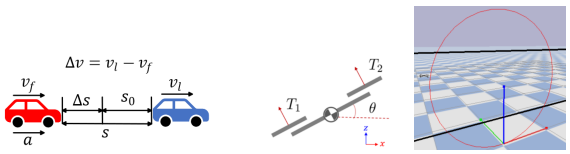


(b) The distribution of states and actions by Q-learning(FPI)



(c) The distribution of states and actions by Q-learning(FRPI)

Figure 5. The policy convergence by Q-learning(Penalty) and FRPI. FRPI pruned the policy space significantly, achieving better experimental results regardless of the environmental dimension.



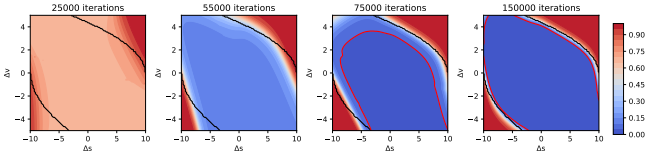
(a) ACC (b) Quadrotor

Figure 6. Classical Environment

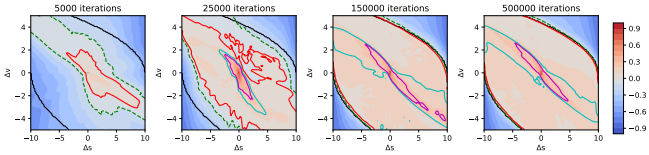
position in a particular range. Fig. 6(b) gives a schematic of this environment. The settings of the system are seen in Appendix D.2.
The result shows that FRPI has a more efficient region expansion speed than FPI. In particular, different color contours indicate that smaller FR regions can be identified, which enable further pruning of state space. Besides, FPI and FRPI achieve the optimal performance in cost and return, while the latter has a faster convergence speed, as shown in Fig. 7(c) and Fig. 8(c).

6.5. Safety Gym Experiment

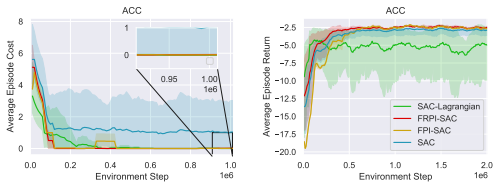
To answer question (3), we compare the algorithms on four high-dimensional robot navigation tasks in Safety Gym (Ray et al., 2019), which are much more complicated and challenging. **PointGoal** and **CarGoal** are two robot navigation tasks, the aims of which are to control the robot (in red) to



(a) The feasible region of ACC by FPI. The red indicates feasible region, which does not appear until **75k** iterations.

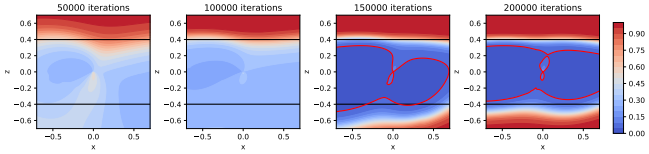


(b) The feasible reachable region of ACC by FRPI. Green dotted line indicates feasible region, which appear at **5k** iterations. Red, cyan, and blood red indicate FR Regions at different γ_g^N values $(0.1, 0.2, 0.3)$.

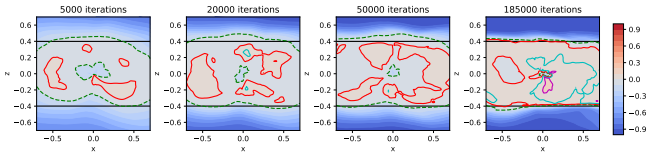


(c) The cost and return of ACC.

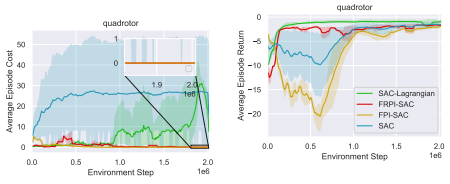
Figure 7. The region-expansion of ACC.



(a) The feasible region of Quadrotor by FPI. The red indicates feasible region, which does not appear until **150k** iterations.



(b) The feasible reachable region of Quadrotor by FRPI. Green dotted line indicates feasible region, which appear at **5k** iterations. Red, cyan, and blood red indicate FR Regions at different γ_g^N values $(0.01, 0.05, 0.1)$.



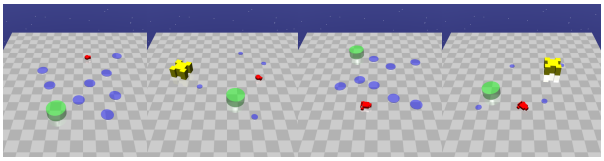
(c) The cost and return of Quadrotor

Figure 8. The region-expansion of Quadrotor.

reach a goal (in green) while avoiding hazards (in blue), as shown in Fig. 9(a) and Fig. 9(c).

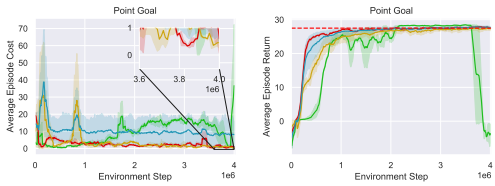
PointPush and **CarPush** except that the robots are trying to push a box (in yellow) to the goal, as shown in Fig. 9(b) and Fig. 9(d). Details of settings can be seen in Appendix D.3.

PointPush and **CarPush** except that the robots are trying to push a box (in yellow) to the goal, as shown in Fig. 9(b) and Fig. 9(d). Details of settings can be seen in Appendix D.3.

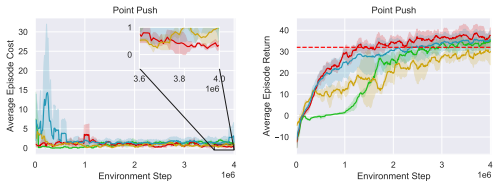


(a) PointGoal (b) PointPush (c) CarGoal (d) CarPush

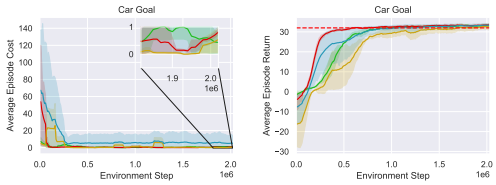
Figure 9. Snapshots of four Safety Gym tasks



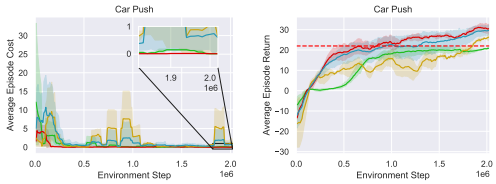
(a) Point Goal



(b) Point Push



(c) Car Goal



(d) Car Push

Figure 10. Optimization results for safety-gym.

As shown in Fig. 10, FRPI-SAC achieves near-zero constraint violations on all tasks, demonstrating low and stable episode cost curves. In comparison, the curves of SAC and SAC-Lag failed to converge to zero or have severe fluctua-

tions. Moreover, our proposed algorithm also exhibits outstanding returns on all the tasks, both in convergence speed, stable training processing, and final performance. Although some non-constraint algorithms like SAC achieve close returns, it comes with sacrificing safety. Appendix Table. 3 shows excellent performance of convergence speed, and Fig. 10 also shows that our method gets much better performance than others with limited data (200k-300k iterations). The time consumption experiment showed that no additional computational burden was added, which is because only one more scenery module, a simple and computationally efficient MLP module, was added to the FRPI framework. As shown in Appendix Table. 1 and Appendix Table. 2, the computational cost of the scenery module update is similar to that of the critic module. Furthermore, benefiting from the further policy space pruning, we achieved better or comparable convergence computer time consumption than SAC.

7. Conlusion

We propose the FRPI for safe RL, where the feasible reachable function can simultaneously identify the reachability and feasibility to achieve efficient policy space pruning. This function naturally satisfies the self-consistent condition and the risky Bellman equation, which can be solved by the fixed point iteration method. On this basis, we propose the feasible reachable policy iteration (FRPI) divided into three steps: policy evaluation, region expansion, and policy improvement. In the region expansion step, the convergence of the feasible region is accelerated, and simultaneously a smaller FR region is identified. The experiment shows the proposed method can identify the feasible region at the start of the training and converge to the maximum feasible reachable region quickly, which is more than three to five times faster than other safe RL approaches on average. Besides, FRPI achieves better performance compared with constraint-free algorithms like SAC with limited data without safety sacrifice, which is significant to real-world application.

The performance of algorithm is best for tasks with naturally clear goal information, like Car Goal or Car Push, etc. However, we should manually build a goal function for some regulating tasks like ACC to guide the agent to approach the target space. The performance of algorithm will depend on the effectiveness of the goal function design. In future work, we will provide an extended definition of feasible reachability to enhance its generalization.

Acknowledgements

This study was supported by National Key R&D Program of China with 2022YFB2502901, National Natural Science Foundation of China (NSFC) under grant number 52221005 and under Grants 52202487. This study was also supported in part by the Young Elite Scientists Sponsorship Program by CAST under Grant 2023QNRC001.

Impact Statement

This paper lays the theoretical groundwork for safe and efficient exploration, a cornerstone in advancing autonomous driving and robotics for safety-critical tasks. Our theoretical contributions establish a robust foundation underlying algorithm development balancing exploration and safety. This is especially relevant for intelligent driving systems and robotics with high costs of errors. The methodologies and insights enhance exploration efficiency in such systems, serving as a beacon for future safety-oriented machine learning research. Our study aims at enabling safety-prioritized innovations in autonomous systems without compromising performance.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

Altman, E. *Constrained Markov decision processes*. Routledge, 2021.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. IEEE, 2019.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Bharadhwaj, H., Kumar, A., Rhinehart, N., Levine, S., Shkurti, F., and Garg, A. Conservative safety critics for exploration. *arXiv preprint arXiv:2010.14497*, 2020.

Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., and Schoellig, A. P. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Chang, Y.-C., Roohi, N., and Gao, S. Neural lyapunov control. *Advances in neural information processing systems*, 32, 2019.

Chen, B., Francis, J., Oh, J., Nyberg, E., and Herbert, S. L. Safe autonomous racing via approximate reachability on ego-vision. *arXiv preprint arXiv:2110.07699*, 2021.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18 (167):1–51, 2018a.

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018b.

Duan, J., Ren, Y., Zhang, F., Li, J., Li, S. E., Guan, Y., and Li, K. Encoding distributional soft actor-critic for autonomous driving in multi-lane scenarios [research frontier][research frontier]. *IEEE Computational Intelligence Magazine*, 19(2):96–112, 2024.

Guan, Y., Ren, Y., Sun, Q., Li, S. E., Ma, H., Duan, J., Dai, Y., and Cheng, B. Integrated decision and control: Toward interpretable and computationally efficient driving intelligence. *IEEE transactions on cybernetics*, 53(2): 859–873, 2022.

Ha, S., Xu, P., Tan, Z., Levine, S., and Tan, J. Learning to walk in the real world with minimal human effort. In *Conference on Robot Learning*, pp. 1110–1120. PMLR, 2021.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hsu, K.-C., Rubies-Royo, V., Tomlin, C. J., and Fisac, J. F. Safety and liveness guarantees through reach-avoid reinforcement learning. *arXiv preprint arXiv:2112.12288*, 2021.

Li, S. E. *Reinforcement learning for sequential decision and optimal control*. Springer, 2023.

Liu, C. and Tomizuka, M. Control in a safe set: Addressing safety in human-robot interactions. In *Dynamic Systems and Control Conference*, volume 46209, pp. V003T42A003. American Society of Mechanical Engineers, 2014.

Luo, Y. and Ma, T. Learning barrier certificates: Towards safe reinforcement learning with zero training-time violations. *Advances in Neural Information Processing Systems*, 34:25621–25632, 2021.

Ma, H., Liu, C., Li, S. E., Zheng, S., and Chen, J. Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning. In *The 4th Annual Learning for Dynamics and Control Conference*, volume 168, pp. 97–109. PMLR, 2022.

Peng, B., Duan, J., Chen, J., Li, S. E., Xie, G., Zhang, C., Guan, Y., Mu, Y., and Sun, E. Model-based chance-constrained reinforcement learning via separated proportional-integral lagrangian. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Ray, A., Achiam, J., and Amodei, D. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7:1, 2019.

Richards, S. M., Berkenkamp, F., and Krause, A. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pp. 466–476. PMLR, 2018.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897. PMLR, 2015.

So, O. and Fan, C. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning. *arXiv preprint arXiv:2305.14154*, 2023.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *International Conference on Learning Representations*, 2019.

Yang, Q., Simão, T. D., Tindemans, S. H., and Spaan, M. T. Wcsac: Worst-case soft actor critic for safety-constrained reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(12), pp. 10639–10646, 2021.

Yang, Q., Simão, T. D., Tindemans, S. H., and Spaan, M. T. Safety-constrained reinforcement learning with a distributional safety critic. *Machine Learning*, 112(3):859–887, 2023a.

Yang, Y., Jiang, Y., Liu, Y., Chen, J., and Li, S. E. Model-free safe reinforcement learning through neural barrier certificate. *IEEE Robotics and Automation Letters*, 2023b.

Yang, Y., Zhang, Y., Zou, W., Chen, J., Yin, Y., and Li, S. E. Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning. *IEEE Transactions on Automatic Control*, 2023c.

Yang, Y., Zheng, Z., and Li, S. E. Feasible policy iteration. *arXiv preprint arXiv:2304.08845*, 2023d.

Yu, D., Ma, H., Li, S., and Chen, J. Reachability constrained reinforcement learning. In *International Conference on Machine Learning*, pp. 25636–25655. PMLR, 2022.

Yuan, Z., Hall, A. W., Zhou, S., Brunke, L., Greeff, M., Panerati, J., and Schoellig, A. P. Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics. *IEEE Robotics and Automation Letters*, 7(4):11142–11149, 2022.

Zheng, Y., Li, J., Yu, D., Yang, Y., Li, S. E., Zhan, X., and Liu, J. Safe offline reinforcement learning with feasibility-guided diffusion model. *arXiv preprint arXiv:2401.10700*, 2024.

A. Feasibility and Reachability

A.1. Feasibility

Definition A.1 (Constrained Set)**.** Let $h : X \to \mathbb{R}$ be a state constraint function. We say that $\mathrm{X}_{\mathrm{cstr}} \triangleq \{x \mid h(x) \leq 0\}$ is a constrained set.

Definition A.2 (Feasible State)**.** A state x is feasible in T steps if there exists a policy π, such that all successive states under π satisfy the state constraints, i.e., $\exists \pi,$ s.t. $x_t \in \mathrm{X}_{\mathrm{cstr}}, t = 0, 1, \ldots, T,$ where $x_0 = x$.

Definition A.3 (Feasible Policy)**.** A policy π is feasible in a state x in T steps if all the successive states under π satisfy the state constraint, i.e., $x_t \in \mathrm{X}_{\mathrm{cstr}}, t = 0, 1, 2, \ldots, T,$ where $x_0 = x$.

Definition A.4 (Feasible Region)**.** The feasible region of π, denoted as $\mathrm{X}^{\pi}_{\mathrm{feas}}$ is the set of all states in which π is feasible. The infeasible region of π is $(\mathrm{X}^{\pi}_{\mathrm{feas}})^c = \mathcal{X} \setminus \mathrm{X}^{\pi}_{\mathrm{feas}}$.

Definition A.5 (Maximum Feasible Region)**.** The maximum feasible region, denoted as $\mathrm{X}^{*}_{\mathrm{feas}}$, is the set of all feasible states. The infeasible region is $(\mathrm{X}^{*}_{\mathrm{feas}})^c = \mathcal{X} \setminus \mathrm{X}^{*}_{\mathrm{feas}}$.

A.2. FR Region Relationship

It can be established that $\mathrm{X}^{*}_{\mathrm{FR}}$ is a subset of the intersection of $\mathrm{X}^{*}_{\mathrm{feas}}$ and $\mathrm{X}^{*}_{\mathrm{reach}}$, denoted as $\mathrm{X}^{*}_{\mathrm{FR}} \subseteq (\mathrm{X}^{*}_{\mathrm{feas}} \cap \mathrm{X}^{*}_{\mathrm{reach}})$. This follows from the existence of policies π_1 and π_2 such that a state x belongs to $\mathrm{X}^{\pi_1}_{\mathrm{feas}}$ under policy π_1 and to $\mathrm{X}^{\pi_2}_{\mathrm{reach}}$ under policy π_2. However, it is possible that π_1 and π_2 are distinct, implying that the state x might not simultaneously satisfy the criteria for being a feasible reachable state under a single policy.

B. Proof

B.1. FR Function

Proof. For all x in $\mathcal{X}$,

$$
\begin{aligned}
& F^{\pi}(x) > 0 \\
\iff & N^{\pi}_{g}(x) < C \in \mathbb{N}^{+}, N^{\pi}_{g}(x) < N^{\pi}_{c}(x) \\
\iff & h(x_t) \leq 0, t = 0, 1, \ldots, T, g(x_T) = 1,
\end{aligned}
$$

where $x_0 = x$ and $\{x_t\}_{t=1}^{T}$ are sampled by π. Therefore, the zero-superlevel set of the FR function is the feasible reachable region of the corresponding policy. Thus, $\{x \in \mathcal{X} | F^{\pi}(x) > 0\} = \mathrm{X}^{\pi}$. On the other hand,

$$
\begin{aligned}
& F^{\pi}(x) = 0 \\
\iff & N^{\pi}_{g}(x) = N^{\pi}_{c}(x) = \infty \\
\iff & h(x_t) \leq 0, t = 0, 1, \ldots, \infty,
\end{aligned}
$$

where $x_0 = x$ and $\{x_t\}_{t=1}^{\infty}$ are sampled by π. The analysis indicates that the feasible region can be denoted as $\{x \in \mathcal{X} | F^{\pi}(x) = 0\} = \mathrm{X}^{\pi}_{feas}$.

B.2. Maximum Feasible Reachable Region

Proof. For all x in $\mathcal{X}$,

$$
\begin{aligned}
& F^{*}(x) > 0 \\
\iff & \exists\, \pi, \text{ s.t. } F^{\pi}(x) > 0 \\
\iff & h(x_t) \leq 0, t = 0, 1, \ldots, T, g(x_T) = 1 \\
\iff & x \text{ is feasible reachable},
\end{aligned}
$$

where $x_0 = x$ and $\{x_t\}_{t=1}^{T}$ are sampled by π.

B.3. Self-Consistency Condition

Proof. As shown in Fig. 2, we divide the state space into four sets: $\mathrm{X_{FR}}$, $\bar{\mathrm{X}}^{\pi}_{\text{feas}} \cap \mathrm{X_{cstr}}$, $\bar{\mathrm{X}}^{\pi} \cap \mathrm{X_{feas}}$ and $\mathrm{X_{cstr}}$. They do not overlap with each other and their union is $\mathcal{X}$. We prove self-consistency condition (4) separately in these four sets.

- $\forall x \in \bar{\mathrm{X}}_{\text{cstr}}$, the constraint is already violated, and thus $c(x) = -1$ and $N_c^{\pi}(x) = 0$. We have $F^{\pi}(x) = -\gamma^0 = -1 = c(x)$, so (4) holds.
- $\forall x \in \bar{\mathrm{X}}^{\pi}_{\text{feas}} \cap \mathrm{X_{cstr}}$, the constraint is not violated, i.e., $c(x) = 0$, but will be violated in a finite number of steps, which satisfies $N_c^{\pi}(x) = N_c^{\pi}(x') + 1$. We have $F^{\pi}(x) = -\gamma^{N_c^{\pi}(x)} = -\gamma \cdot \gamma^{N_c^{\pi}(x')} = -\gamma F^{\pi}(x')$, and thus (4) holds.
- $\forall x \in \mathrm{X}^{\pi}$, the constraint is not violated and will reach the target in the finite horizon, which means $c(x) = 0$ and $N_g^{\pi}(x) = T$. The next state x' is still in $\mathcal{X}^{\pi}$, so $N_g^{\pi}(x') = T - 1$. We have $F^{\pi}(x) = \gamma^T = \gamma\gamma^{T-1} = \gamma F^{\pi}(x')$, and thus (4) holds.
- $\forall x \in \bar{\mathrm{X}}^{\pi} \cap \mathrm{X_{feas}}$, the goal is not will be reached and constraint will never be violated in the infinite horizon, which means $g(x) = 0$ and $N_g^{\pi}(x) = \infty$. The next state x' is still in the set, so $N_g^{\pi}(x') = \infty$. We have $F^{\pi}(x) = \gamma^{\infty} = 0 = F^{\pi}(x')$, and thus (4) holds. In conclusion, $\forall x \in \mathrm{X}$, (4) holds.

B.4. Unique Fixed Point

Proof. Consider the metric space (M, d_∞) with $M = \{F \mid F : X \to [0, 1]\}$ and d_∞ being the uniform metric. First, we prove that (M, d_∞) is complete. Let $\{F_n\}$ be any Cauchy sequence in M, then

$$\forall \varepsilon > 0, \exists N \geq 1, \text{ s.t. } m, k \geq N, d_\infty(F_m, F_k) < \varepsilon,$$

which means

$$|F_m(x) - F_k(x)| \leq \|F_m - F_k\|\|x\| < \varepsilon, \quad \forall x \in \mathcal{X}. \quad (*)$$

Hence, for all $x \in X$, $\{F_n(x)\}$ is a Cauchy sequence in $([0, 1], d_\infty)$, which is a complete space (since $[0, 1]$ is a closed subset of $\mathbb{R}$). It follows that $\{F_n(x)\}$ is a convergent sequence for all $x \in X$. Let $F_n(x) \to F(x) \in [0, 1]$. Apparently, $F \in M$. Hold m and let $k \to \infty$ in $(*)$, then we have

$$|F_m(x) - F(x)| < \varepsilon, \forall x \in X,$$

which is equivalent to

$$d_\infty(F_m, F) < \varepsilon.$$

Thus we have $F_n \to F \in M$, so (M, d_∞) is complete.

Then, we prove that D^{π} is a contraction mapping on (M, d_∞). For all x in X

$$\begin{aligned}
&|D^{\pi}F_1(x) - D^{\pi}F_2(x)| \\
&= g(x) + c(x) + (1 - g(x))((1 + c(x)) \\
&\quad \gamma(F_1(x') - F_2(x')) \\
&\leq \gamma|F_1(x') - F_2(x')| \\
&\leq \gamma d_\infty(F_1, F_2).
\end{aligned}$$

Hence,

$$\begin{aligned}
d_\infty(D^{\pi}F_1, D^{\pi}F_2) &= \sup_x |D^{\pi}F_1(x) - D^{\pi}F_2(x)| \\
&\leq \gamma d_\infty(F_1, F_2).
\end{aligned}$$

Since $\gamma \in (0, 1)$, D^{π} is a contraction mapping. Acoording to Banach's fixed-point theroem, D^{π} has a unique fixed point. Note that F^{π} is a fixed point of D^{π}, the unique fixed point is F^{π}.

B.5. FR Region Expansion

We assume that the FR function and the state-value function can be accurately approximated and study the relationship of FRPI in two adjacent iterations.

Proof. According to (5), for all x in X^{π_k}, the condition $F^{\pi_k}(x') > 0$ implies $x' \in X^{\pi_k}$, where $x' = f(x, \pi_{k+1}(x))$. This implication confirms that the set X^{π_k} is forward invariant. Consequently, it follows that for every x in X^{π_k}, the sequence $x, x', x'', \ldots$ remains within X^{π_k}. This ensures that for $x \in X^*$, we have $x' \in X^*, x'' \in X^*, \ldots$. Therefore, the control $u = \pi_{k+1}(x)$ belongs to the control set U^{π_k}, which is a subset of U^*, leading to the conclusion that π_{k+1} is an optimal policy belonging to Π^*.

Furthermore, the inequality

$$r(x, \pi_{k+1}(x)) + \gamma V^{\pi_k}(f(x, \pi_{k+1}(x))) \geq \quad r(x, \pi_k(x)) + \gamma V^{\pi_k}(f(x, \pi_k(x)))$$

holds true, which is equal to

$$\begin{aligned} V^{\pi_k}(x) &\leq r_0 + \gamma V^{\pi_k}(x_1) \\ &\leq r_0 + \gamma r_1 + \gamma^2 V^{\pi_k}(x_2) \\ &\vdots \\ &\leq r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \\ &= \sum_{t=0}^{T} \gamma^t r_t \\ &= V^{\pi_{k+1}}(x). \end{aligned}$$

We assume that $N_g^{\pi_k} \geq N_g^{\pi_{k+1}}$ when policy is improved in goal-reaching problem, which is equal to $\gamma^{N_g^{\pi_k}} \leq \gamma^{N_g^{\pi_{k+1}}}$, so $F^{\pi_{k+1}}(x) \geq F^{\pi_k}(x) > 0$ is satisfied. Hence, we deduce that $x \in X^{\pi_{k+1}}$ must be true. According to (6), for all x in $(X^{\pi_k})^c$,

$$\begin{aligned} & F^{\pi_k}(f(x, \pi_{k+1}(x)) \\ &= \max_u F^{\pi_k}(f(x, u)) \\ &\geq F^{\pi_k}(f(x, \pi_k(x)). \end{aligned}$$

Thus, we have

$$\begin{aligned} & F^{\pi_k}(x) \\ &= g_x + c_x + (1 - g_x)(1 + c_x)\gamma F^{\pi_k}(f(x, \pi_k(x)) \\ &\leq g_x + c_x + (1 - g_x)(1 + c_x)\gamma F^{\pi_k}(f(x, \pi_{k+1}(x)). \end{aligned}$$

Let $\{x_t\}_{t=0}^{\infty}$ be the state sequence in a trajectory under π_{k+1}, where $x_0 = x$. We denote $c(x_t)$ as c_t for simplicity. We have

$$\begin{aligned} & F^{\pi_k}(x) \\ &\leq g_0 + c_0 + (1 - g_0)(1 + c_0)\gamma F^{\pi_k}(x_1) \\ &\leq g_0 + c_0 + (1 - g_0)(1 + c_0)\gamma \\ &\quad (g_1 + c_1 + (1 - g_1)(1 + c_1)\gamma F^{\pi_k}(x_2)) \\ &\leq g_0 + c_0 + \gamma(1 - g_0)(1 + c_0)(g_1 + c_1) \\ &\quad + \gamma^2(1 - g_0)(1 + c_0)(1 - g_1)(1 + c_1)(g_2 + c_2) \\ &\vdots \\ &= \sum_{t=0}^{T} \gamma^t \prod_{s=0}^{t-1}(1 - g_s)(1 + c_s)(g_t + c_t) \\ &= \gamma^{N^{\pi_{k+1}}(x)}(g_T + c_T) \\ &= F^{\pi_{k+1}}(x), \end{aligned}$$

which shows the region expansion tendency of FRPI out of the region. We will prove the convergence of expansion in the next section.

B.6. Region-wise Policy Improvement

Proof. For all x in X^{π_k}, we have $\pi_k(x) \in U^{\pi_k}(x)$. According to (5), we have $\pi_{k+1}(x) \in U^{\pi_k}(x)$ and

$$\begin{aligned} V^{\pi_k}(x) &= r(x, \pi_k(x)) + \gamma V^{\pi_k}(f(x, \pi_k(x))) \\ &\leq \max_{u \in \mathrm{U}^{\pi_k}(x)} r(x, u) + \gamma V^{\pi_k}(f(x, u)) \\ &= r(x, \pi_{k+1}(x)) + \gamma V^{\pi_k}(f(x, \pi_{k+1}(x))). \end{aligned}$$

Let $\{x_t\}_{t=0}^{\infty}$ be the state sequence in a trajectory under π_{k+1}, where $x_0 = x$. We denote $r(x_t, \pi_{k+1}(x_t))$ as r_t for simplicity.

$$\begin{aligned} V^{\pi_k}(x) &\leq r_0 + \gamma V^{\pi_k}(x_1) \\ &\leq r_0 + \gamma r_1 + \gamma^2 V^{\pi_k}(x_2) \\ &\vdots \\ &\leq r_0 + \gamma r_1 + \gamma^2 r_2 + \dots \\ &= \sum_{t=0}^{\infty} \gamma^t r_t \\ &= V^{\pi_{k+1}}(x). \end{aligned}$$

B.7. Feasible Reachable Bellman Equation

The state-value function $V : X^\star \to \mathbb{R}$ is the optimal state-value function V^* if and only if it satisfies the feasible Bellman equation for all x in X^*

$$V(x) = \max_{u \in \mathrm{U}^*(x)} r(x, u) + \gamma V(x'), \tag{10}$$

First, we prove that the optimal state-value function satisfies the feasible Bellman equation. For all $x \in \mathcal{X}^*$,

$$\begin{aligned} V^*(x) &= \max_{\pi \in \Pi^*} V^\pi(x) \\ &= \max_{u \in \mathrm{U}^*} \left\{ \sum_{t=0}^{T} \gamma^t r(x_t, u_t) \right\} \\ &= \max_{u \in \mathrm{U}^*} \left\{ r(x, u) + \max_{u \in \mathrm{U}^*} \left\{ \sum_{t=1}^{T} \gamma^t r(x_t, u_t) \right\} \right\} \\ &= \max_{u \in \mathrm{U}^*} \{r(x, u)\} + \gamma \max_{\pi \in \Pi^*} V^\pi(x') \\ &= \max_{u \in \mathrm{U}^*} \{r(x, u)\} + \gamma V^*(x'). \end{aligned}$$

Then, we prove that the feasible Bellman operator B^*, which is defined as

$$(B^*V)(x) = \max_{u \in \mathrm{U}^*} \{r(x, u) + \gamma V(x')\},$$

has a unique fixed point. Consider a metric space (M, d_∞) with $M = \{V | V : \mathcal{X}^* \to \mathbb{R}\}$. The proof of its completeness is similar to that of $\{\mathcal{F} | \mathcal{F} : \mathcal{X} \to [0, 1]\}, d_\infty$ and hence omitted here. We only prove that B^* is a contraction mapping on it. For $V_1, V_2 \in M$ and $x \in \mathcal{X}^*$, define

$$u_1 = \arg\max_{u \in \mathrm{U}^*} \{r(x, u) + \gamma V_1(x')\},$$

$$u_2 = \arg\max_{u \in \mathrm{U}^*} \{r(x,u) + \gamma V_2(x')\},$$

and let $r_1 = r(x,u_1)$, $r_2 = r(x,u_2)$, $x_1' = f(x,u_1)$, $x_2' = f(x,u_2)$. Then, it follows that

$$\begin{aligned} B^*V_1(x) - B^*V_2(x) &= r_1 + \gamma V_1(x_1') - (r_2 + \gamma V_2(x_2')) \\ &\le r_1 + \gamma V_1(x_1') - (r_1 + \gamma V_2(x_1')) \\ &= \gamma(V_1(x_1') - V_2(x_1')). \end{aligned}$$

Similarly, we have

$$B^*V_2(x) - B^*V_1(x) \le \gamma(V_2(x_2') - V_1(x_2')).$$

Define

$$z(x) = \max\{|\gamma(V_1(x_1') - V_2(x_1'))|, |\gamma(V_2(x_2') - V_1(x_2'))|\}.$$

We have

$$|B^*V_1(x) - B^*V_2(x)| \le z(x).$$

Since

$$\begin{aligned} z(x) &\le \gamma \max\{|V_1(x_1') - V_2(x_1')|, |V_2(x_2') - V_1(x_2')|\} \\ &\le \gamma d_\infty(V_1, V_2), \end{aligned}$$

we have

$$|B^*V_1(x) - B^*V_2(x)| \le \gamma d_\infty(V_1, V_2),$$

and

$$d_\infty(B^*V_1(x), B^*V_2(x)) \le \gamma d_\infty(V_1, V_2).$$

Thus, B^* is a contraction mapping on (M, d_∞). According to Banach's fixed-point theorem, B^* has a unique fixed point, which is V^*. Therefore, V^* is the unique solution to the feasible Bellman equation. Thus, the feasible Bellman equation is a necessary and sufficient condition of the optimal state-value function.

B.8. Risky Bellman equation

The FR function $F\colon \mathcal{X} \to \mathbb{R}$ is the optimal FR function if and only if it satisfies the risky Bellman equation for all x in $\mathcal{X}$

$$F(x) = c(x) + g(x) + (1 - g(x))(1 - c(x))\gamma \max_u F(x').$$

Proof. First, we prove that the optimal FR satisfies the risky Bellman equation. We divide the state space into four sets: X_{FR}, $\bar{\mathrm{X}}^\pi_{\text{feas}} \cap \mathrm{X}_{\text{cstr}}$, $\bar{\mathrm{X}}^\pi \cap \mathrm{X}_{\text{feas}}$ and X_{cstr}. They do not overlap with each other and their union is X.

- $\forall x \in \bar{\mathrm{X}}_{\text{cstr}}$, the constraint is already violated, and thus $c(x) = -1$ and $N_c^\pi(x) = 0$. We have $F^*(x) = -\gamma^0 = -1 = c(x)$, so (9) holds.
- $\forall x \in (\bar{\mathrm{X}}^\pi_{\text{feas}} \cap \mathrm{X}_{\text{cstr}})$, the constraint is not violated currently, i.e., $c(x) = 0$ and $\exists \pi$, s.t. $F^\pi(x) = F^*(x)$. Then, $\max_u F^*(x') = \max_u F^\pi(x') = F^\pi(f(x,\pi(x)))$, since $F^\pi(x) = \gamma F^\pi(f(x,\pi(x)))$, we have $F^*(x) = \gamma \max_u F^*(x')$, thus (9) holds.
- $\forall x \in \mathrm{X}^\pi$, the constraint is not violated and will reach the target in the finite horizon, which means $c(x) = 0$, $N_g^\pi(x) = T$, and $\exists \pi$, s.t. $F^\pi(x) = F^*(x)$. Then, $\max_u F^*(x') = \max_u F^\pi(x') = F^\pi(f(x,\pi(x)))$, since $F^\pi(x) = \gamma F^\pi(f(x,\pi(x)))$, we have $F^*(x) = \gamma \max_u F^*(x')$, thus (9) holds.
- $\forall x \in (\bar{\mathrm{X}}^\pi \cap \mathrm{X}_{\text{feas}})$, the goal is not will be reached and constraint will never be violated in the infinite horizon, which means $g(x) = 0$ and $N_g^\pi(x) = \infty$. The next state x' is still in the set, so $N_g^\pi(x') = \infty$. We have $F^\pi(x) = \gamma^\infty = 0 = F^\pi(x')$, $F^*(x) = \max_u F^*(x') = 0$ and thus (9) holds. In conclusion, $\forall x \in \mathrm{X}$, (9) holds.

□

Then, we prove that the right-hand side of (9) is a contraction mapping under the uniform metric. Define the risky Bellman operator D^* as

$$(D^*F)(x) = c(x) + g(x) + (1 - g(x))(1 + c(x))\gamma \max_u F(x').$$

Proof. For all FR functions F_1, F_2 and for all state x in $\mathcal{X}$, let $u_1 = \arg\max_u F_1(x'), u_2 = \arg\max_u F_2(x')$, and $x_1' = f(x, u_1), x_2' = f(x, u_2)$, then

$$\begin{aligned} & D^*F_1(x) - D^*F_2(x) \\ &= (1 + c(x))(1 - g(x))\gamma(F_1(x_1') - F_2(x_2')) \\ &\leq (1 + c(x))(1 - g(x))\gamma(F_1(x_2') - F_2(x_2')). \end{aligned}$$

Similarly, we have

$$D^*F_2(x) - D^*F_1(x) \leq (1 - g_x)(1 + c_x)\gamma(F_2(x_1') - F_1(x_1')).$$

Define

$$z(x) = \max\left\{\left|(1 + c_x)(1 - g_x)\gamma(F_1(x_1') - F_2(x_1'))\right|, \left|(1 + c_x)(1 - g_x)\gamma(F_2(x_2') - F_1(x_2'))\right|\right\}.$$

We have

$$|D^*F_1(x) - D^*F_2(x)| \leq z(x).$$

Since

$$\begin{aligned} z(x) &\leq \gamma \max\{|F_1(x_1') - F_2(x_1')|, |F_2(x_2') - F_1(x_2')|\} \\ &\leq \gamma d_\infty(F_1, F_2), \end{aligned}$$

we have

$$|D^*F_1(x) - D^*F_2(x)| \leq \gamma d_\infty(F_1, F_2),$$

and

$$d_\infty(D^*F_1(x), D^*F_2(x)) \leq \gamma d_\infty(F_1, F_2).$$

Thus, D^* is a contraction mapping. Together with the completeness of $(F \mid F\colon \mathcal{X} \to [-1, 1]\}, d_\infty)$ proved before, according to Banach's fixed-point theorem, D^* has a unique fixed point, which is F^*. Therefore, F^* is the unique solution to the risky Bellman equation. Thus, the risky Bellman equation is a necessary and sufficient condition of the optimal FR function.

B.9. Convergence of FRPI

Suppose at the k-th iteration, for all x in $\mathcal{X}$, $F^{\pi_{k+1}}(x) = F^{\pi_k}(x)$, and for all x in X^*, $V^{\pi_{k+1}}(x) = V^{\pi_k}(x)$. In the proof, we denote $g(x)$ as g_x, $c(x)$ as c_x for simplicity.

Proof. First, we prove that F^{π_k} is the solution to the risky Bellman equation. According to (5), for all x in X^{π_k},

$$F^{\pi_k}(f(x, \pi_{k+1}(x))) \geq 0.$$

According to (6), for all x in $(X^{\pi_k})^c$,

$$F^{\pi_k}(f(x, \pi_{k+1}(x))) = \max_u F^{\pi_k}(f(x, u)).$$

We have

$$\begin{aligned} & F^{\pi_{k+1}}(x) \\ &= c_x + g_x + (1 - g_x)(1 + c_x)\gamma F^{\pi_{k+1}}(f(x, \pi_{k+1}(x))) \\ &= c_x + g_x + (1 - g_x)(1 + c_x)\gamma F^{\pi_k}(f(x, \pi_{k+1}(x))) \\ &= c_x + g_x + (1 - g_x)(1 + c_x)\gamma \max_u F^{\pi_k}(f(x, u)) \\ &= c_x + g_x + (1 - g_x)(1 + c_x)\gamma \max_u F^{\pi_{k+1}}(f(x, u)). \end{aligned}$$

Thus, $F^{\pi_{k+1}}$ is the solution to the risky Bellman equation. Since $F^{\pi_k} = F^{\pi_{k+1}}$, F^{π_k} is also the solution to the risky Bellman equation.

Next, we prove that V^{π_k} is the solution to the feasible Bellman equation. Since $F^{\pi_k} = F^*$, $\mathrm{U}^{\pi_k}(x) = \{u \in U | x' \in \mathrm{X}^*\} = \mathrm{U}^*(x)$. We have

$$\begin{aligned} V^{\pi_{k+1}}(x) &= r(x, \pi_{k+1}(x)) + \gamma V^{\pi_{k+1}}(f(x, \pi_{k+1}(x))) \\ &= r(x, \pi_{k+1}(x)) + \gamma V^{\pi_k}(f(x, \pi_{k+1}(x))) \\ &= \max_{u \in \mathrm{U}^{\pi_k}(x)} r(x, u) + \gamma V^{\pi_k}(x') \\ &= \max_{u \in \mathrm{U}^*(x)} r(x, u) + \gamma V^{\pi_{k+1}}(x'). \end{aligned}$$

Thus, $V^{\pi_{k+1}}$ is the solution to the feasible Bellman equation. Since $V^{\pi_k} = V^{\pi_{k+1}}$, V^{π_k} is also the solution to the feasible Bellman equation.

Thus, both F^{π_k} and V^{π_k} are optimal, i.e., $F^{\pi_k} = F^*$ and $V^{\pi_k} = V^*$. Because in finite state and action spaces, the number of policies is finite, this process converges to the maximum feasible region and the optimal state-value function in a finite number of iterations.

C. Practical Implementation

In this subsection, we introduce some practical implementations of feasible reachable policy iteration (FRPI). We first discuss some techniques when approximating the FR function with a neural network in infinite state spaces. We then show how to combine FRPI with a mainstream RL algorithm, soft actor-critic (SAC) (Haarnoja et al., 2018) , to yield a practical safe RL algorithm.

C.1. Approximation of FR function

To deal with infinite state spaces, we use a neural network with a tanh output activation function to approximate the FR function, which incorporate the knowledge that the value of FR function is between -1 and 1 into the training process. In feasible region identification, we optimize the FR network to approximate the FR function of the current policy. Therefore, we use the right-hand side of the self-consistency condition (4) as the training label, i.e.

$$L_F(\phi) = -\mathbb{E}\left\{y_F \log F_\phi(x) + (1 - y_F) \log(1 - F_\phi(x))\right\}, \tag{11}$$

where

$$y_F = c(x) + g(x) + (1 + c(x))(1 - g(x))\gamma F_\phi(x').$$

C.2. Integration with SAC

We combine FRPI with soft actor-critic (SAC) (Haarnoja et al., 2018) and denote the resulting algorithm as FRPI-SAC. It learns an action FR function network G_ϕ, two Q networks $Q_{\omega_1}, Q_{\omega_2}$, and a policy network π_θ. The action FR function network G takes the current state and action as input and outputs the FR function value of the next state, i.e., $G(x, u) = F(f(x, u))$. Its loss function is

$$L_G(\phi) = -\mathbb{E}_{(x,u)\sim D}\left\{y_G \log G_\phi(x, u) + (1 - y_G) \log(1 - G_\phi(x, u))\right\}, \tag{12}$$

where

$$y_G = c(x) + g(x) + (1 - g(x))(1 + c(x))\gamma G_\phi(x', u'),$$

and ϕ is the parameters of the target action FR function network, which is updated slower than the action FR function network for stabilizing the training process.

The loss functions of the Q networks are,

$$L_Q(\omega_i) = \mathbb{E}_{(x,u,r,x')\sim D}\left\{(y_Q - Q_{\omega_i}(x, u))^2\right\}, \tag{13}$$

where

$$y_Q = r + \gamma\left(\min_{j\in\{1,2\}} Q_{\omega_j}(x', u') - \alpha \log \pi_\theta(u' \mid x')\right), \tag{14}$$

where $i \in \{1, 2\}$, ω_j are the parameters of the target Q networks, and α is the temperature.

The policy loss in a feasible state is

$$l_f(x) = l_r(x) - \frac{1}{t} \cdot \log(G_\phi(x, u)), \tag{15}$$

$$l_r(x) = \alpha \log \pi_\theta(u \mid x) - \min_{i \in \{1,2\}} Q_{\omega_i}(x, u), \tag{16}$$

where $u \sim \pi_\theta(\cdot \mid x)$.

The policy loss in an state out of feasible reachable region is

$$l_o(x) = G_\phi(x, u). \tag{17}$$

The total policy loss is

$$L_\pi(\theta) = \mathbb{E}_{x \sim D} \left\{ m(x) l_f(x) + (1 - m(x)) l_o(x) \right\}, \tag{18}$$

where $m(x) = 1$ if $G_\phi(x, u) > 0$.

The loss function of the temperature is

$$L(\alpha) = \mathbb{E}_{x \sim D} \left\{ -\alpha \log \pi_\theta(u \mid x) - \alpha \mathcal{H} \right\}, \tag{19}$$

where $\mathcal{H}$ is the target entropy.

D. Environment

D.1. ACC

Both following and leading vehicles are modeled as point masses moving in a straight line.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \mathbf{u}, \tag{20}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \triangleq \begin{bmatrix} \Delta s & \Delta v \end{bmatrix}^T$, with $\Delta s = s - s_0$ standing for the difference between actual distance s and expected distance s_0 between the two vehicles and Δv standing for the relative velocity. The action $\mathbf{u} \triangleq \begin{bmatrix} a \end{bmatrix}$ is the acceleration of the following vehicle.

The reward function is defined as

$$r(\mathbf{x}, \mathbf{u}) = -0.001 \Delta s^2 - 0.01 \Delta v^2 - a^2, \tag{21}$$

and $r = 1$ if the $(|x_1| \leq 0.1)$ and $(|x_2| \leq 0.1)$
and the constraint function is

$$h(\mathbf{x}) = |\Delta s| - \Delta s_{\text{max}}, \tag{22}$$

which, since a large acceleration is penalized in (21), will be violated by a performance-only policy.

The ACC task is a regulating task, not a typical goal-reaching task. The quality of the construction of the goal function affects the stability of its return. In this problem, we construct a goal function based on the region around the regulating point, and the quality of the region design affects the variance of its return. Since the return function of ACC depends on the distance from the regulating point, which is limited by the design of the goal function, the return will have a relatively large variance in the early stage of learning. At the same time, it can be seen that the variance of return will decrease significantly after its policy is improved when the agent can steadily approach the regulating point.

D.2. Quadrotor

$$\mathbf{x} \triangleq \begin{bmatrix} x & z & \dot{x} & \dot{z} & \theta & \dot{\theta} \end{bmatrix}^T,$$

where (x, z) is the position of the quadrotor on xz-plane, and θ is the pitch angle. The action of the system is

$$\mathbf{u} \triangleq \begin{bmatrix} T_1 & T_2 \end{bmatrix}^T,$$

including the thrusts generated by two pairs of motors.

The goal is to minimize the tracking error with minimal efforts:

$$r(\mathbf{x}, \mathbf{u}) = -\left\| \begin{bmatrix} x - x_{\text{ref}} & z - z_{\text{ref}} \end{bmatrix} \right\|^2 - 0.1\dot{\theta}^2 - \theta^2 - 0.1(T_1 - T_0)^2 - 0.1(T_2 - T_0)^2, \tag{23}$$

and $r = 1$ if $\left\| \begin{bmatrix} x - x_{\text{ref}} & z - z_{\text{ref}} \end{bmatrix} \right\|^2 \leq 0.001$
where $(x_{\text{ref}}, z_{\text{ref}})$ is the reference position the quadrotor is supposed to be at, and T_0 is the thrust needed for balancing the gravity. The reference position moves along a circle $x^2 + z^2 = 0.5^2$ with a constant angular velocity, but the constraint function is

$$h(\mathbf{x}) = \begin{bmatrix} |z| - z_{\max} \\ |\theta| - \theta_{\max} \\ \left\| \begin{bmatrix} x - x_{\text{ref}} & z - z_{\text{ref}} \end{bmatrix} \right\| - err_{\max} \end{bmatrix} \leq 0, \tag{24}$$

restricting the quadrotor to stay in a rectangular area.

D.3. Safety Gym

PointGoal and **CarGoal** are two robot navigation tasks, the aims of which are to control the robot (in red) to reach a goal (in green) while avoiding hazards (in blue), as shown in 9 (a) and 9 (C). There are eight hazards with a radius of 0.2 and a goal with a radius of 0.3. The state includes the robot's velocity, the goal's position, and LiDAR point clouds of the hazards. The control inputs of the robots are the torques of their motors, controlling the motion of moving forward and turning for a Point robot, and the left and right wheels for a Car robot.
PointPush and **CarPush** except that the robots are trying to push a box (in yellow) to the goal, as shown in Fig. 9 (b) and Fig. 9 (d). There are four hazards with a radius of 0.1 and a goal with a radius of 0.3. The state further includes the position of the box.Details of settings followed the (Yang et al., 2023d).
We conducted training on an NVIDIA GPU 3090 using JAX, setting XLA_PYTHON_CLIENT_MEM_FRACTION to 0.1, which allocates 2720 MB of GPU memory. The inference times for various algorithms on Safety Gym are shown in Table 1. FRPI-SAC and FPI-SAC demonstrate similar inference times, with FRPI-SAC at 1.576 ms and FPI-SAC at 1.573 ms. RAC shows an inference time of 1.589 ms, while SAC-Lag and SAC exhibit longer times of 1.742 ms and 0.983 ms, respectively. Table 2 presents the convergence speeds on Safety Gym across different tasks measured in million iterations.

Table 1. The Inference Time on Safety Gym (ms)

Algorithm	**FRPI-SAC**	**FPI-SAC**	**RAC**	**SAC-Lag**	**SAC**
Inference Time (ms)	1.576	1.573	1.589	1.742	0.983

Table 2. The Convergence Speed on Safety Gym (million iterations)

Convergence time(s)	**PointGoal**	**PointPush**	**CarGoal**	**CarPush**
FRPI-SAC	**2.0K**	**4.5k**	**2.2K**	**8.6k**
SAC	2.5K	4.6k	2.3K	9.3k
SAC-Lag	7.3K	15.2k	5.5K	30.0k
FPI-SAC	6.2K	12.3k	5.3K	20.7k
RAC	4.8K	8.3k	5.6K	14.2k

Note:Training on an NVIDIA GPU 3090 using JAX.
Note:XLA_PYTHON_CLIENT_MEM_FRACTION=0.1 (2720MB GPU).

We added the safe certificate of the HJ method, RAC[2], which integrates the SAC and RCRL, as a supplement to the baseline. The result showed that RAC realizes zero constraint violation as other safe RL baselines, but the sample efficiency still needs to be improved. Reachability analysis emphasizes the absolute guarantee of safety. However, the inaccuracy of early safety certificates in the early stage also leads to excessive pruning of state space, which indirectly leads to low sample efficiency in the early optimization process.

Table 3. Performance Comparison of Algorithms

Algorithm	PointGoal	PointPush	CarGoal	CarPush
FRPI-SAC	**1.10 ± 0.05**	**1.20 ± 0.10**	**0.50 ± 0.05**	**0.65 ± 0.06**
SAC	1.50 ± 0.08	2.20 ± 0.10	0.80 ± 0.05	0.90 ± 0.10
SAC-Lag	2.20 ± 0.20	2.80 ± 0.30	1.00 ± 0.15	2.00 ± 0.16
FPI-SAC	3.20 ± 0.10	4.00 ± 0.20	1.60 ± 0.08	1.80 ± 0.05
RAC	3.00 ± 0.10	2.30 ± 0.10	1.00 ± 0.10	1.95 ± 0.15

D.4. Hyperparameters

The hyperparameters used in the experiments are listed in Tab. 4.

Table 4. Hyperparameters of the algorithms

Hyperparameter	Classic | Safety Gym
Shared	
Discount factor	0.99
Number of hidden layers	2
Number of hidden neurons	256
Optimizer(Adam)	$(\beta_1 = 0.99, \beta_2 = 0.999)$
SAC-related	
Activation function	ReLU
Target entropy	$-\dim(\mathcal{U})$
Initial temperature	1.0
Target smoothing coefficient	0.005
Learning rate	1e-4
Batch size	256 | 1024
Replay buffer size	2×10^6 | 4×10^6
Lagrange-related	
Initial Lagrange multiplier	1.0
Multiplier learning rate	1e-4
Multiplier delay	10
FPI-related	
Feasibility threshold (ρ)	≤ 0.05
Initial t	1.0
t increase factor	1.1
t update delay	10000
FRPI-related (ours)	
Feasible reachable threshold (ρ)	> 0
Initial t	1.0
t increase factor	1.1
t update delay	10000